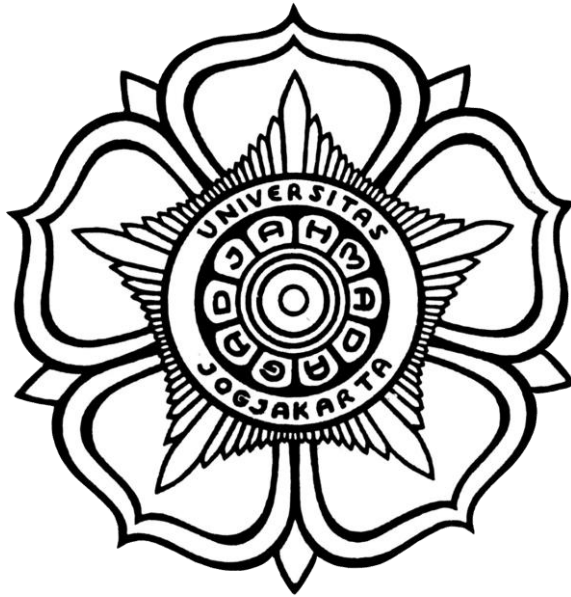


**PENGEMBANGAN APLIKASI PENGINGAT TENGGAT WAKTU  
PEMBAYARAN HUTANG SECARA OTOMATIS BERBASIS  
WEBSITE PT. HEXINDO ADIPERKASA TBK.**

**SKRIPSI**



**Disusun oleh:**

**HAFIZH ABDURRACHMAN MA'AS**

**15/385395/TK/44057**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI  
INFORMASI  
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA  
YOGYAKARTA**

**2019**

## **HALAMAN PENGESAHAN**

# **PENGEMBANGAN APLIKASI PENGINGAT TENGGAT WAKTU PEMBAYARAN HUTANG SECARA OTOMATIS BERBASIS WEBSITE PT. HEXINDO ADIPERKASA TBK.**

## **SKRIPSI**

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana Teknik Program S-1  
Pada Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik  
Universitas Gadjah Mada

Disusun oleh:

**HAFIZH ABDURRACHMAN MA'AS**

**15/385395/TK/44057**

Telah disetujui dan disahkan  
pada tanggal 9 Mei 2019

Dosen Pembimbing I

Dosen Pembimbing II

**Warsun Najib, S.T., M.Sc.**

**NIP 197311251998031003**

**Adhistya Erna Permanasari, S.T.,  
M.T., Ph.D.**

**NIP 198104292008122001**

## PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini:

Nama : Hafizh Abdurrachman Ma'as  
NIM : 15/385395/TK/44057  
Tahun terdaftar : 2015  
Program Studi : Teknologi Informasi  
Fakultas : Teknik Universitas Gadjah Mada

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 9 Mei 2019

Materai Rp 6.000  
(Tanda tangan)

Hafizh Abdurrachman Ma'as  
15/385395/TK/44057

## **HALAMAN PERSEMBAHAN**

Tugas akhir ini kupersembahkan kepada Ayah dan Ibu yang selalu membimbing,  
memotivasi, memberikan kasih sayang sepanjang masa kuliah.  
Kupersembahkan pula kepada Bang Rafdi dan Bang Fauzan yang selalu mendukung dan  
menyediakan segala kebutuhan, dan membantu menyelesaikan setiap masalah sepanjang  
masa kuliah.

Kepada Riztya, dan Kak Bila yang selalu memberikam semangat dan hiburan,  
Serta teman-teman semua yang selalu memeberikan pengalaman, cerita, canda, dan tawa,  
semoga kita semua selalu diberikan kemudahan, keberkahan, dan rahmatNya.

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Sarjiya, S.T., M.T., Ph.D., selaku Ketua Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada.
2. Hanung Adi Nugroho, S.T., M.E., Ph.D., selaku Ketua Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada.
3. Bapak Warsun Najib, S.T., M.Sc., selaku dosen pembimbing pertama yang telah memberikan arahan, bimbingan, dukungan, inspirasi, dan saran kepada penulis dalam menyelesaikan tugas akhir ini.
4. Ibu Adhistya Erna Permanasari, S.T., M.T., Ph.D., selaku dosen pembimbing kedua yang telah memberikan bimbingan, dukungan, inspirasi, dan saran kepada penulis dalam menyelesaikan tugas akhir ini.
5. Bapak Ir. Sujoko Sumaryono, M.T., selaku dosen pembimbing akademik yang telah memberikan arahan, bimbingan, dukungan, dan motivasi kepada penulis dalam menjalani kehidupan perkuliahan terutama bagian akademik.
6. Bapak dan Ibu Dosen serta karyawan DTETI FT UGM yang selalu bekerja keras untuk memberikan yang terbaik untuk para generasi penerus bangsa.
7. Kedua orang tua penulis, Bapak Norsopen Ma'as dan Ibu Kristiani yang tidak kenal lelah untuk mencari nafkah, memberikan kasih sayang, semangat, dan motivasi yang tiada tara sehingga penulis dapat menyelesaikan perkuliahan ini dan dapat menjadi seorang sarjana teknik. Semoga kalian selalu diberikan kesehatan, kemudahan, keberkahan, dan rahmat dariNya.
8. Kedua abang penulis, Rafdi Milzano Ma'as dan Moehammad Dzaky Fauzan Ma'as yang selalu membantu penulis baik dalam menyediakan segala kebutuhan kuliah dan mencari solusi ketika penulis menghadapi suatu masalah dalam kegiatan perkuliahan. Semoga kalian selalu diberikan kesehatan, kemudahan, kesuksesan, keberkahan, dan rahmat dariNya.

9. Adik serta Kakak Ipar penulis, Riztya Khairunnisa Ma'as dan Isma Nabila yang selalu memberikan semangat serta hiburan kepada penulis. Semoga Riztya diterima di Fakultas dan Universitas yang diinginkan dan menjadi kebanggaan orang tua, serta Kak Bila dimudahkan dalam segala urusannya.
10. Mas Hendra yang sudah menjembatani dari pihak PT. Hexindo Adiperkasa, Tbk. dalam hal berkomunikasi kepada penulis sehingga proyek ini dapat dijadikan sebagai judul tugas akhir penulis.
11. Seluruh tim Pantau-AR PT.Hexindo Adiperkasa, Tbk. yang telah membantu menyediakan segala kebutuhan penulis dalam pengerjaan proyek yang dijadikan sebagai tugas akhir ini.
12. Ahmad Irsyad Fitra , Ahmad Thariq Syauqi, Biesta Adri Azizi, Muhammad Ashil Allathief, Muhammad Edward Chakim, Nanda Brilianto, Nugraha Aji Swara, dan Nuwas Dzarrin Tantowi yang selalu memberikan pengalaman, berbagi hal baru, cerita, masukan, dan menemani berbagai kegiatan lainnya. Semoga kalian selalu diberikan kemudahan dalam menghadapi segala urusan, kesuksesan dalam karir, dan rahmat dariNya.
13. Teman-teman IT B yang selalu memberikan hal baru dan menarik, serta sering mengadakan jalan-jalan bersama ke beberapa tempat. Semoga kalian selalu diberikan kemudahan dalam segala urusan serta kesuksesan dalam menggapai cita-cita kalian.
14. Semua pengurus Divisi Workshop KMTETI UGM yang memberikan pengalaman berharganya dalam berorganisasi.
15. Kepada seluruh teman-teman Teknik Elektro dan Teknologi Informasi 2015. Terima kasih atas tawa, canda, sedih, sudah, cerita, dan pengalaman yang sudah kalian bagi. Semuanya akan selalu terekam dalam ingatan penulis.

Akhir kata penulis berharap semoga skripsi ini dapat memberikan manfaat bagi penulism kita semua, serta bagi orang banyak, aamiin.

## DAFTAR ISI

HALAMAN PENGESAHAN .....	ii
HALAMAN PERSEMBAHAN .....	iv
KATA PENGANTAR.....	v
DAFTAR ISI .....	vii
DAFTAR TABEL .....	x
DAFTAR GAMBAR.....	xi
DAFTAR SINGKATAN.....	xiii
INTISARI.....	xiv
ABSTRACT .....	xii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Tugas akhir .....	3
1.4 Tujuan Tugas akhir .....	3
1.5 Manfaat Tugas akhir .....	3
1.6 Sistematika Penulisan .....	4
1.6.1. Bab I : PENDAHULUAN .....	4
1.6.2. Bab II : TINJAUAN PUSTAKA DAN DASAR TEORI.....	4
1.6.3. Bab III : METODE TUGAS AKHIR .....	4
1.6.4. Bab IV : HASIL DAN PEMBAHASAN.....	4
1.6.5. Bab V : KESIMPULAN .....	4
BAB II TINJAUAN PUSTAKA DAN DASAR TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Dasar Teori.....	8
2.2.1. Hypertext Markup Language .....	8

2.2.2. Cascading Style Sheet.....	8
2.2.3. JavaScript.....	8
2.2.4. Structured Query Language .....	9
2.2.5. PHP Hypertext Preprocessor .....	9
2.2.6. CodeIgniter .....	9
2.2.7. Sistem Penjadwalan .....	10
2.2.8. Rapid Application Development (RAD) .....	12
2.2.9. Aplikasi Berbasis Website .....	14
2.2.10. Domain and Hosting .....	14
2.2.11. Black Box Testing .....	16
<b>BAB III METODE PENELITIAN .....</b>	<b>18</b>
3.1 Alat dan Bahan.....	18
3.1.1. Alat.....	18
3.1.2. Bahan .....	18
3.2 Alur Tugas akhir .....	19
3.2.1. Identifikasi Masalah.....	19
3.2.2. Studi Literatur .....	20
3.2.3. Analisis Kebutuhan.....	20
3.2.4. Perancangan Aplikasi .....	21
3.2.5. Pengembangan Aplikasi .....	26
3.2.6. Pengujian Aplikasi.....	27
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>30</b>
4.1 Implementasi basis data .....	30
4.2 Pengujian tiap fungsi dengan metode <i>black box</i> .....	32
4.2.1. Fitur login dan multilevel user.....	32
4.2.2. Fitur Transaction List (Supervisor dan Seller) .....	37
4.2.3. Fitur Edit Data .....	38



4.2.4. Fitur Paid Off (Supervisor dan Seller) .....	40
4.2.5. Fitur Input Data Manual (Supervisor dan Seller) .....	41
4.2.6. Fitur Input Data Excel.....	43
4.2.7. Fitur Export to PDF .....	45
4.2.8. Fitur <i>Dashboard</i> .....	46
4.2.9. Fitur Change Password .....	47
4.2.10. Fitur Menambah Akun Baru .....	48
4.2.11. Fitur Reminder Otomatis .....	50
4.2.12. Evaluasi hasil pengujian black box.....	50
4.3 Pengingat Tenggat Waktu Otomatis .....	54
4.3.1. Pengaturan Cron Job pengingat tenggat waktu otomatis.....	54
4.3.2. Script back end fungsi pengingat tenggat waktu otomatis .....	55
4.3.3. Tampilan email pengingat tenggat waktu otomatis .....	60
4.4 Perbandingan sistem baru terhadap sistem lama .....	61
4.4.1. Kelebihan Sistem .....	61
4.4.2. Kekurangan Sistem .....	61
4.4.3. Perbedaan cara kerja sistem baru dan sistem lama .....	62
BAB V KESIMPULAN DAN SARAN .....	63
5.1 Kesimpulan .....	63
5.2 Saran .....	64
DAFTAR PUSTAKA.....	65
LAMPIRAN .....	67

## DAFTAR TABEL

Tabel 2.1 Analisis Perbandingan Metode.....	7
Tabel 3.1 Daftar Fitur yang dibutuhkan pihak PT. Hexindo Adiperkasa Tbk. ....	20
Tabel 3.2 Daftar Fitur yang akan dibuat.....	26
Tabel 3.3 Daftar Kriteria Pengujian seluruh fitur.....	28
Tabel 4.1 Tabel <i>user</i> .....	30
Tabel 4.2 Tabel penjualan .....	31
Tabel 4.3 Hasil pengujian seluruh fitur dengan metode <i>black box</i> .....	51
Tabel 4.4 Perbedaan sistem baru dan sistem lama .....	62

## DAFTAR GAMBAR

Gambar 2.1 Alur kerja MVC.....	10
Gambar 2.2 Cara penulisan waktu pada <i>cron job</i> .....	11
Gambar 2.3 Alur Rapid Application Development (RAD) [13] .....	14
Gambar 2.4 Metode <i>Black Box Testing</i> .....	16
Gambar 3.1 Alur Metode Penelitian.....	19
Gambar 3.2 <i>Use Case Diagram</i> seluruh <i>Role</i> .....	22
Gambar 3.3 Algoritme Pengingat Pembayaran Otomatis .....	23
Gambar 4.1 Hasil uji coba <i>login</i> dengan menggunakan <i>username</i> dan <i>password</i> admin .....	33
Gambar 4.2 Hasil uji coba <i>login</i> dengan menggunakan <i>username</i> dan <i>password supervisor</i> .....	34
Gambar 4.3 Hasil uji coba <i>login</i> dengan menggunakan <i>username</i> dan <i>password seller</i> .....	35
Gambar 4.4 Hasil uji coba <i>login</i> dengan menggunakan <i>username</i> atau <i>password</i> yang salah.....	36
Gambar 4.5 Tampilan transaction list seller (atas) dan supervisor (bawah) .....	37
Gambar 4.6 Tampilan fitur <i>edit</i> data pada akun <i>supervisor</i> .....	39
Gambar 4.7 Tampilan <i>paid off seller</i> (atas) dan <i>supervisor</i> (bawah) .....	40
Gambar 4.8 Tampilan penginputan data secara manual.....	42
Gambar 4.9 Contoh penginputan data dengan <i>file excel</i> .....	43
Gambar 4.10 Tampilan <i>Upload file excel</i> .....	44
Gambar 4.11 Tampilan <i>preview file excel</i> .....	44
Gambar 4.12 Tampilan berhasil memasukkan data dari <i>file excel</i> (atas) dan <i>transaction list</i> setelah proses <i>input</i> data berhasil (bawah) .....	45
Gambar 4.13 Tampilan <i>export PDF</i> .....	46
Gambar 4.14 Tampilan Daftar Barang .....	47
Gambar 4.15 Tampilan <i>change password</i> (atas) dan <i>pop up</i> berhasil mengubah <i>password</i> .....	48
Gambar 4.16 Tampilan menambahkan akun baru (atas) dan <i>pop up</i> berhasil menambahkan akun baru .....	49
Gambar 4.17 Tampilan akun baru melakukan <i>login</i> .....	50
Gambar 4.18 Konfigurasi <i>cron job</i> .....	54
Gambar 4.19 <i>Script code reminder</i> bagian 1 .....	55

Gambar 4.20 <i>Script code reminder</i> bagian 2.....	56
Gambar 4.21 <i>Script code reminder</i> bagian 3.....	57
Gambar 4.22 <i>Script code body email reminder</i> .....	59
Gambar 4.23 Tampilan <i>email reminder</i> yang diterima oleh pengguna.....	60

## DAFTAR SINGKATAN

### **C**

CSS	<i>Cascading Style Sheet</i>
CMS	<i>Content Management System</i>

### **H**

HTML	<i>Hypertext Markup Language</i>
------	----------------------------------

### **M**

MVC	<i>Model, View, Controller</i>
-----	--------------------------------

### **P**

PHP	<i>PHP Hypertext Preprocessor</i>
-----	-----------------------------------

### **R**

RAD	<i>Rapid Application Development</i>
-----	--------------------------------------

### **S**

SMS	<i>Short Message Service</i>
SQL	<i>Structured Query Language</i>
SDLC	<i>Software Development Life Cycle</i>

### **T**

TOP	<i>Term Of Payment</i>
-----	------------------------

## INTISARI

PT. Hexindo Adiperkasa, Tbk. merupakan salah satu perusahaan penyedia alat berat terkemuka di Indonesia. Pemilik saham terbesar sekaligus sebagai induk dari perusahaan ini adalah Hitachi Construction Machinery Co. Ltd yang didirikan di Jepang. Perusahaan ini menyediakan layanan penjualan serta penyewaan alat berat, namun pencatatan *customer* masih dilakukan secara terpusat dan pengisian data dilakukan dengan cara manual menggunakan Microsoft Excel. Selain hal tersebut, proses pemberian informasi waktu tenggat pembayaran tagihan masih dilakukan secara manual dengan menggunakan data pada Microsoft Excel.

Oleh karena itu, dibutuhkan sebuah aplikasi berbasis *website* untuk memudahkan *user* dalam penyimpanan data *customer* dengan lengkap dan saling terintegrasi. Selain itu, diperlukan fitur pengingat pembayaran tagihan kepada *customer* secara otomatis dengan menggunakan sistem penjadwalan.

Dengan adanya aplikasi ini diharapkan *user* tidak perlu lagi memberikan informasi mengenai tenggat waktu pembayaran tagihan secara manual dan mampu meningkatkan efektivitas dan efisiensi waktu serta sumber daya yang diperlukan *user* baik dari sisi *seller* maupun *supervisor* dalam meningkatkan pelayanannya terhadap *customer*.

Kata kunci : Sistem Informasi, Website, Pengingat, Pembayaran, Hutang

## ABSTRACT

*PT. Hexindo Adiperkasa, Tbk. Is one of the leading provider of heavy equipment in Indonesia. The biggest share owner and parent of this company is Hitachi Construction Machinery Co. Ltd established in Japan. This company provides heavy equipment sales and rental services, but recording customers is still done centrally and data collection is done manually by using Microsoft Excel. In addition to this, the process of providing information during the deadline for payment of bills is still done manually using data in Microsoft Excel.*

*Therefore, a website-based application is needed to make it easier for users to store customer data completely and mutually integrated. In addition, a reminder feature is required for automatic customer bill payments using a scheduling system.*

*With this application it is expected that users no longer need to provide information about the deadline for billing payments manually and are able to increase the effectiveness and efficiency of time and resources needed by users both from the seller's side and supervisors in improving their service to the customer.*

*Keywords : Information System, Website, Reminder, Payment, Debt*

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

PT. Hexindo Adiperkasa, Tbk. Merupakan salah satu perusahaan penyedia alat berat terkemuka di Indonesia [1]. Mulai dari mini excavator dengan berbagai ukuran (kecil, medium, dan besar), truk, sampai berbagai macam *sparepart* alat berat sekaligus penyediaan jasa *service*. Sejak pertama berdiri sampai sekarang, perusahaan ini sudah memiliki banyak pelanggan (*customer*), baik dari sisi penjualan maupun sisi layanan.

Dengan jumlah pelanggan yang banyak, sistem informasi memiliki peran penting dalam proses pencatatan, penyimpanan, serta pengolahan data tersebut, namun sampai saat ini sistem informasi yang digunakan memiliki banyak kekurangan. Pencatatan data pelanggan tersebut dilakukan secara manual, yaitu dengan cara pengguna (*seller / supervisor*) menginputkan data pada *file* penyimpanan utama dalam bentuk Microsoft Excel, sehingga data-data pelanggan dapat terhimpun menjadi terpusat. Setiap hari pihak administrator melakukan pengecekan terhadap *file* penyimpanan utama guna mengetahui pelanggan mana saja yang tenggat waktu pembayarannya pembayarannya sudah dekat. Setelah itu, admin mengirimkan *email* yang berisikan informasi tagihan pembayaran kepada pelanggan yang bersangkutan.

Cara kerja di atas memiliki 2 kendala, seperti sistem informasi yang digunakan masih bersifat *offline*, sehingga akan sangat susah untuk diakses secara *remote*. Hal ini akan berdampak pada kesulitan dalam proses pengintegrasian data, di mana data harus dikirimkan ke lokasi tempat *file* master (*file excel*) berada. Lalu kendala yang kedua adalah pengguna harus selalu memeriksa *file excel* setiap hari guna mengetahui pelanggan mana saja yang harus dikirimkan *email* mengenai informasi tagihan pembayaran. Dibutuhkan ketelitian dan kesabaran dalam mengerjakan proses ini agar tidak ada pelanggan yang terlewat dalam pengiriman email pengingat.

Selain kedua kendala di atas, perusahaan juga memiliki kendala terhadap hak akses *file* master (*file excel*) pada masing-masing *role user* (*supervisor* dan *seller*). Seharusnya hanya *supervisor* yang mampu mengelola seluruh data *customer* (menambah, mengubah, dan menghapus data). *Seller* hanya mampu mengelola data pelanggan yang membeli maupun konsultasi melalui perantara dirinya (tidak dapat mengelola data pelanggan *seller* lain), namun karena *file* penyimpanannya masih dalam bentuk *file excel*, maka pengelolaan data pelanggan mampu dilakukan oleh siapapun yang memiliki akses ke *file* master.



Dengan alasan-alasan di atas, pihak perusahaan meminta untuk dibuatkan aplikasi berbasis *website* yang mampu mencatat seluruh data pembayaran, pengingat bagi *customer* secara otomatis, dan pembagian hak akses berdasarkan *role user (multilevel user)* dengan desain yang sederhana tetapi menarik dan mudah untuk digunakan. Selain itu, perusahaan juga membutuhkan beberapa fitur tambahan seperti *dashboard* serta pengolahan data dengan format keluaran pdf guna memudahkan proses pengerjaannya.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang diuraikan di atas, diperoleh rumusan masalah untuk tugas akhir sebagai berikut:

- a. Sistem pencatatan *customer* yang digunakan PT. Hexindo Adiperkasa, Tbk masih dalam bentuk Microsoft Excel, sehingga penambahan, pengurangan, maupun pengubahan data hanya dapat dilakukan oleh satu orang yang memiliki *file* utamanya saja. Dari Tugas Akhir ini, solusi yang dibangun berdasarkan *requirement* dari pihak PT. Hexindo Adiperkasa, Tbk. adalah dengan membuat aplikasi berbasis website yang mampu melakukan pencatatan dan manipulasi (*create, read, update, dan delete*) data *customer* secara *online* dan flexibel. Selain hal tersebut, data-data *customer* tersimpan secara tersentralisasi di suatu database yang terletak pada *cloud hosting*. Hal ini dilakukan untuk meminimalisir adanya replikasi data. Dengan adanya fitur ini, diharapkan mampu membantu memudahkan dan memaksimalkan pekerjaan karyawan PT. Hexindo Adiperkasa, Tbk.
- b. Pengingat tagihan pembayaran kepada *customer* masih dilakukan secara manual, sehingga membutuhkan karyawan untuk melakukan pemantauan data tenggat waktu dan mengirimkan *email* pengingat pembayaran kepada *customer*. Dari Tugas Akhir ini, solusi yang dibangun berdasarkan *requirement* dari pihak PT. Hexindo Adiperkasa, Tbk. adalah dengan membuat fitur pengingat otomatis berbasis *email* kepada para *customer* yang diatur dengan sistem penjadwalan. Dengan adanya fitur ini, karyawan tidak perlu lagi mengingatkan waktu tenggat pembayaran kepada *customer* secara manual, melainkan cukup dengan memasukkan data *customer* kedalam aplikasi.

- c. Pihak perusahaan juga membutuhkan pembagian hak akses ke sistem informasi mereka agar terdapat batasan yang jelas mengenai hal apa saja yang boleh dilakukan oleh *seller* maupun *supervisor*. Selain itu perusahaan juga membutuhkan katalog produk yang mereka tawarkan dan pengolahan data dengan format keluaran pdf. Dengan Tugas Akhir ini, solusi yang dibangun berdasarkan *requirement* dari pihak PT. Hexindo Adiperkasa, Tbk. adalah dengan membuat fitur pengenalan produk pada suatu halaman khusus, membuat pembatasan hak akses aplikasi level karyawan, dan fitur cetak data berbaas pdf. Dengan fitur ini diharapkan mampu meningkatkan performa pekerjaan dari karyawan.

### **1.3 Batasan Tugas akhir**

Aplikasi yang dikembangkan pada Tugas Akhir ini memiliki batasan dalam beberapa aspek. Batasan tersebut diantaranya :

1. Data yang digunakan pada Tugas Akhir ini bersifat data *dummy*, karena aplikasi ini dikembangkan untuk internal perusahaan sehingga pihak PT. Hexindo Adiperkasa Tbk tidak memberikan akses untuk data asli mereka.
2. Sistem penjadwalan *reminder* otomatis akan terus berjalan selama server *hosting* aktif. Apabila server *hosting* sedang mengalami gangguan, maka sistem penjadwalan *reminder* otomatis pun akan tidak bekerja.

### **1.4 Tujuan Tugas akhir**

Tujuan dilaksanakan tugas akhir ini adalah:

1. Mengembangkan sistem informasi pencatatan data pelanggan yang sudah ada menjadi *online* dan dapat saling terintegrasi.
2. Menyediakan fitur pengingat tagihan pembayaran secara otomatis.
3. Menyediakan fitur pengenalan produk, *multilevel user*, serta pengolahan data dengan format keluaran pdf untuk memaksimalkan sistem informasi yang ada.

### **1.5 Manfaat Tugas akhir**

Manfaat dari tugas akhir ini adalah :

1. Sistem informasi semakin mudah untuk diakses secara *remote* dan data *customer* saling terintegrasi.
2. Mengurangi sumberdaya dan waktu yang digunakan administrator dalam mengirimkan *email* yang berisikan tagihan pembayaran kepada *customer*.

3. Mewadahi pekerjaan-pekerjaan manual lainnya menjadi terpusat guna memudahkan pengguna dalam pengolahan data.

## **1.6 Sistematika Penulisan**

Dalam penulisan laporan tugas akhir ini, penulis menggunakan sistematika untuk menjelaskan keseluruhan isi dari pengembangan aplikasi. Adapun sistematika penulisan adalah sebagai berikut :

### **1.6.1. Bab I : PENDAHULUAN**

Pada bab ini berisi mengenai pengantar dan pengenalan terhadap masalah yang akan dipecahkan melalui pengembangan suatu aplikasi oleh penulis. Secara rinci bab ini berisikan latar belakang, rumusan masalah, batasan tugas akhir, tujuan tugas akhir, manfaat tugas akhir, dan sistematika penulisan.

### **1.6.2. Bab II : TINJAUAN PUSTAKA DAN DASAR TEORI**

Pada bab ini berisi mengenai tinjauan pustaka dari beberapa pengembangan aplikasi yang serupa untuk dijadikan landasan dan perbandingan dalam pengembangan aplikasi tugas akhir ini. Selain itu, juga terdapat dasar teori dari komponen-komponen (bahasa pemrograman, sistem penjadwalan, dan *software*) yang digunakan pada pengembangan aplikasi tugas akhir ini.

### **1.6.3. Bab III : METODE TUGAS AKHIR**

Pada bab ini berisi mengenai langkah-langkah yang dilakukan penulis dalam melakukan pengembangan aplikasi tugas akhir ini mulai dari bertemu pihak PT. Hexindo Adiperkasa Tbk untuk menentukan spesifikasi *requirement* sampai dengan melakukan pengembangan aplikasi tersebut. Secara rinci bab ini berisi alat dan bahan tugas akhir dan alur tugas akhir.

### **1.6.4. Bab IV : HASIL DAN PEMBAHASAN**

Pada bab ini berisi penjelasan mengenai aplikasi yang dikembangkan oleh penulis

### **1.6.5. Bab V : KESIMPULAN**

Pada bab ini berisi kesimpulan mengenai pemecahan pada rumusan masalah dengan memberikan solusi pengembangan aplikasi yang dikaitkan dengan tujuan pengerjaan tugas akhir ini. Selain itu terdapat juga saran dalam melakukan pengembangan selanjutnya pada aplikasi ini agar lebih optimal.

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Di era pesatnya perkembangan teknologi saat ini, tak bisa dipungkiri bahwa sudah banyak pekerjaan yang sebelumnya dikerjakan oleh manusia, sekarang sudah digantikan oleh mesin. Hal ini juga sudah dilakukan oleh beberapa perusahaan, di mana mereka mulai menggunakan teknologi otomatis seperti robot, AI, cloud computing, maupun sistem penjadwalan dalam melakukan suatu pekerjaan. Teknologi otomatis ini akan melakukan perintah logika pemrograman dalam pengambilan suatu keputusan, kemampuan mengontrol, serta pengoprasian suatu pekerjaan. Salah satu contoh teknologi otomatis yang banyak digunakan oleh beberapa perusahaan adalah sistem penjadwalan otomatis.

Penjadwalan otomatis merupakan suatu sistem yang mampu melaksanakan perintah-perintah tertentu secara otomatis. Pengguna hanya perlu mengatur kapan dan perintah apa yang akan di jalankan oleh *scheduler* sekali, lalu *scheduler* akan menjalankan perintah tersebut secara berulang, sesuai dengan waktu yang ditentukan diawal. Sistem penjadwalan digunakan karena memiliki banyak kelebihan dalam mengerjakan suatu pekerjaan dibandingkan apabila pekerjaan tersebut dikerjakan oleh manusia. Beberapa keuntungan diantaranya adalah sistem penjadwalan dapat berjalan secara otomatis (hanya perlu diatur diawal), bisa melakukan pekerjaan kapanpun (baik tengah malam ataupun pagi buta), sangat sedikit kemungkinan terjadi kesalahan, dan dapat berjalan terus menerus tanpa ada penambahan hal-hal lainnya. Dengan beberapa keuntungan tersebut, sistem penjadwalan banyak dipilih oleh perusahaan karena dapat meningkatkan kinerja perusahaan tanpa perlu mengeluarkan biaya yang besar.

Penelitian mengenai sistem penjadwalan otomatis telah dilakukan oleh Hans Pratyaksa. Hans Pratyaksa menggunakan sistem penjadwalan otomatis untuk menentukan kapan suatu perangkat elektronik menyala dan kapan perangkat elektronik mati. Penelitian yang dilakukan oleh Hans Pratyaksa menunjukkan bahwa dengan menggunakan sistem penjadwalan secara otomatis mampu menghemat penggunaan listrik serta mengefektifkan tenaga yang digunakan [2]. Aplikasi yang digunakan pada penelitian Hans Pratyaksa adalah Plugwise yang diintegrasikan dengan Google Calender. Google Calender bertugas dalam melakukan reminder tanggal penggunaan perangkat elektronik, sedangkan Plugwise berfungsi dalam pemetaan jadwal rutin.

Penelitian mengenai sistem pendajwalan otomatis lainnya juga sudah dilakukan oleh Hernowo. Hernowo menggunakan *SMS gateway* dengan bantuan Gammu dalam proses *reminder* otomatis mengenai servis rutin kendaraan bermotor kepada pelanggan *dealer* Yamaha Cemara *Cycles* Boyolai [3]. Dengan adanya *SMS gateway* ini dipercayai mampu membuat proses bisnis perusahaan menjadi lebih efisien.

Penelitian selanjutnya mengenai penjadwalan otomatis juga dilakukan oleh Gunawan. Gunawan memilih menggunakan *SMS gateway* dan *email* dalam *reminder* jadwal *check up* di Laboratorium Prima Medika Kudus [4].

Sistem *reminder* otomatis dipilih oleh beberapa perusahaan, lembaga, maupun *client* karena mampu memberikan suatu informasi yang lengkap kepada *user* atau *customer* secara maksimal, efisien, dan praktis [5]. Sehingga dengan adanya hal ini mampu meningkatkan produktivitas kinerja suatu perusahaan atau lembaga tersebut.

Aplikasi sistem *reminder* yang dikembangkan ini masih memiliki kekurangan. Pertama aplikasi ini masih bergantung kepada suatu *device*, sebagai contoh adalah plugwise yang hanya dapat bekerja pada perangkat elektronik dan *SMS gateway* (Gammu) yang hanya dapat bekerja apabila terdapat modem yang selalu terkoneksi ke komputer server selama 24 jam. Kedua aplikasi ini masih bersifat semi otomatis karena untuk pengiriman *email* dan SMS, pihak administrasi harus menginputkan isi pesan dan alamat *email* atau nomor telepon yang di tuju sehingga memerlukan waktu tambahan untuk mengirimkan *reminder* kepada *user* ataupun *customer*.

Tabel 2.1 Analisis Perbandingan Metode

Perbandingan Metode					
Menggunakan plugwise dan google calendar		Menggunakan SMS <i>Gateway</i> dan Gammu		Menggunakan <i>Email</i> melalui <i>Cronjob</i>	
Keuntungan	Kerugian	Keuntungan	Kerugian	Keuntungan	Kerugian
Sangat <i>support</i> untuk reminder yang berbasis alat elektronik (seperti suhu ruangan, solar panel, suhu CPU, dan lain-lainnya).	Bersifat aplikasi yang terintegrasi ke suatu <i>device</i> , sehingga tidak mampu mengirimkan <i>reminder</i> ke pengguna tanpa alat perantara tambahan	Pengguna tidak memerlukan akses internet untuk mendapatkan <i>reminder</i> .	Harus mencolokkan modem yang terkoneksi ke internet ke komputer <i>server</i> selama 24 jam penuh agar sms <i>gateway</i> dapat bekerja maximal	Tidak memerlukan biaya tambahan untuk vendor SMS <i>gateway</i> maupun alat elektronik tertentu (karena sudah ada pada salah satu fasilitas hosting).	Pengguna harus memiliki akses internet agar mendapatkan <i>reminder</i> .
	<i>Device</i> pengirim <i>reminder</i> harus terkoneksi ke jaringan yang sama dengan alat elektronik yang digunakan.		Harus mengeluarkan biaya tambahan untuk vendor apabila tidak ingin menggunakan modem tambahan.	Kapasitas <i>email</i> pengirim dapat diatur sesuai dengan kapasitas memori yang disediakan hosting.	
			<i>Full support</i> untuk aplikasi berbasis <i>website</i> (sudah disediakan <i>email</i> , server, maupun <i>scheduler / cronjob</i> ).		

## 2.2 Dasar Teori

### 2.2.1. Hypertext Markup Language

*Hypertext Markup Language* (HTML) merupakan kode yang berfungsi untuk menyusun struktur konten pada halaman *website*. HTML terdiri dari serangkaian elemen, yang digunakan untuk melampirkan atau membungkus berbagai konten lalu menampilkannya dengan cara tertentu. Misalnya, konten dapat disusun dalam bentuk daftar poin–poin, dalam satu paragraf, atau menggunakan tabel maupun gambar. HTML juga memiliki beberapa *tag* yang berfungsi untuk membuat tulisan miring, menebalkan tulisan, *hyperlink* atau menuju halaman lain, membesar atau mengecilkan ukuran tulisan, dan masih banyak lainnya [6].

### 2.2.2. Cascading Style Sheet

*Cascading Style Sheet* (CSS) merupakan bahasa yang digunakan oleh seorang pengembang web untuk menghias atau memperindah tampilan *website*. CSS mampu menentukan gaya teks, ukuran suatu tabel, tata letak konten, pewarnaan konten, dan pemilihan *font* tulisan [7]. Selain itu css juga dapat memungkinkan untuk menyesuaikan tampilan suatu *website* ke berbagai jenis perangkat , seperti layar besar, layar kecil, maupun printer. CSS dapat dibilang sebagai pelengkap HTML dalam pengembangan dasar suatu *website*. Walaupun demikian, CSS tidak tergantung pada HTML. CSS juga dapat digunakan dalam bahasa markup apapun berbasis XML.

### 2.2.3. JavaScript

*JavaScript* merupakan bahasa *Client Side Programming Language* yang berarti tidak memerlukan proses *compile* karena bahasa ini berjalan atau di proses di sisi *user* / pengguna. Untuk menjalankan *JavaScript*, kita hanya membutuhkan aplikasi *text editor* dan *web browser*. *JavaScript* pada awal perkembangannya berfungsi untuk membuat interaksi antara *user* dengan situs web menjadi lebih cepat tanpa harus menunggu pemrosesan di web server. Namun, dengan seiring perkembangan teknologi, fungsi penggunaan *JavaScript* bertambah menjadi mampu membuat tampilan *website* lebih interaktif dan dinamis. Javascript mampu membuat suatu halaman untuk bereaksi terhadap suatu peristiwa, menunjukkan efek khusus, menerima teks variable, mamvalidasi data, membuat *cookie*, mendeteksi browser yang sedang mengakses laman, dan masih banyak lainnya [8]. Dalam penggunaannya, biasanya *JavaScript* ditanam di dalam kode HTML.

#### 2.2.4. Structured Query Language

*Structured Query Language* (SQL) merupakan bahasa pemrograman khusus yang digunakan untuk memanipulasi (menambah, membaca, mengubah, dan menghapus) data dalam *Relational Database Management System* (RDBMS) [9]. Basis data yang kami gunakan adalah MySQL yang bersifat *open source*. MySQL merupakan aplikasi RDBMS paling populer di dunia, karena selain memiliki kekuatan komputasi yang handal dan cepat, MySQL juga memiliki *sintaks* yang mudah untuk dimengerti.

#### 2.2.5. PHP Hypertext Preprocessor

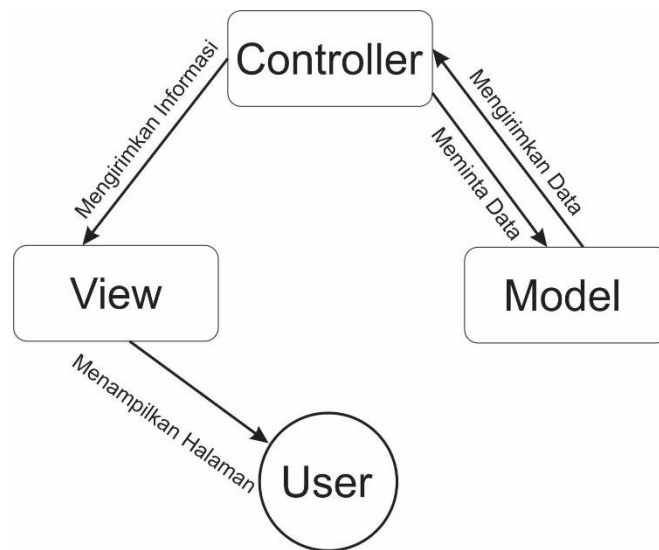
PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. *PHP Hypertext Preprocessor* merupakan salah satu bahasa *back end* pada pemrograman aplikasi berbasis *website*. PHP disebut bahasa pemrograman *server side* karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada web browser (client). Namun, walaupun berjalan pada sisi server, kode PHP dapat dipasang di tengah-tengah kode HTML. Kode PHP disertakan dalam instruksi pemrosesan awal dan akhir khusus `<?php` dan `?>` yang memungkinkan *developer* untuk masuk dan keluar dari "mode PHP" [10].

#### 2.2.6. CodeIgniter

*Framework* (kerangka kerja) merupakan kumpulan dari fungsi-fungsi dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang *programmer*, tanpa harus membuat fungsi atau *class* dari awal. *CodeIgniter* adalah *framework* PHP (*PHP Hypertext Pre processor*) yang kuat dengan ukuran *file* kerangka yang sangat kecil [11]. Walaupun ukuran *file* ini kecil, *file* ini memiliki fungsi-fungsi yang lengkap dan mudah digunakan. *CodeIgniter* bersifat *opensource* sehingga bebas digunakan dalam pengembangan atau pembuatan aplikasi berbasis *website*. Selain ringan dan cepat, *CodeIgniter* juga memiliki dokumentasi yang lengkap dan disertai dengan contoh implementasi kodenya. *CodeIgniter* memiliki prinsip kerja MVC (*Model, View, Controller*). Secara garis besar, konsep MVC ini membagi komponen utama menjadi 3 bagian yaitu *model*, *view*, dan *controller*. *Model* merupakan bagian yang berhubungan secara langsung dalam proses manipulasi (membuat, membaca, mengubah, dan menghapus) data pada basis data. *View* merupakan bagian yang menangani tampilan (*user interface*) halaman yang muncul pada pengguna. Hasil pengolahan data dari *model* akan ditampilkan melalui bagian ini. Lalu yang terakhir adalah *controller*. *Controller* berisikan kumpulan instruksi berbagai proses yang



berfungsi sebagai penghubung antara *model* dengan *view*. Hal ini dilakukan guna menghindari pengaksesan *model* secara langsung oleh pengguna melalui *view*.



Gambar 2.1 Alur kerja MVC

### 2.2.7. Sistem Penjadwalan

Sistem Penjadwalan merupakan suatu layanan yang terdapat pada suatu sistem operasi atau *cloud hosting* untuk menjalankan atau mengeksekusi suatu perintah dalam waktu yang telah ditentukan. Sistem penjadwalan juga biasa dikenal dengan penjadwalan pekerjaan. Sistem penjadwal memulai dan menangani pekerjaan secara otomatis dengan memanipulasi algoritme pekerjaan yang telah diatur oleh pengguna.

Pada saat ini beberapa sistem penjadwalan telah menawarkan antarmuka pengguna grafis . Antarmuka berfungsi agar memudahkan pengguna dalam mengatur waktu dan fungsi mana saja yang akan dieksekusi. [12]

#### 2.2.7.1. Cron Job

*Cron Job* merupakan suatu program penjadwalan yang berfungsi untuk menjalankan atau mengeksekusi suatu perintah secara rutin dengan waktu tertentu. Pada awalnya *cron job* dikembangkan pada sistem operasi Linux, namun saat ini *cron job* juga sudah ada pada sistem operasi windows. Terdapat *daemon cron* yang berjalan pada sistem ini. *Daemon* adalah program yang berjalan di latar belakang setiap saat, biasanya diinisiasikan oleh sistem. *Daemon cron* ini bertanggung jawab

untuk meluncurkan pekerjaan *cron* sesuai jadwal. Jadwal berada di *file* konfigurasi bernama *crontab*. Di sinilah semua tugas dan penghitung waktu terdaftar.

*Cron Job* dijalankan di server, biasanya sudah tersedia di dalam *cloud hosting*. *Cron Job* paling umum digunakan untuk otomatisasi pemeliharaan atau administrasi sistem, namun mereka juga relevan dengan pengembangan aplikasi berbasis *website*. Ada banyak situasi ketika aplikasi berbasis *website* mungkin memerlukan tugas-tugas tertentu untuk dijalankan secara berkala, seperti sistem pengingat otomatis.



Gambar 2.2 Cara penulisan waktu pada *cron job*

*Cron Job* memiliki tata cara penulisan sendiri dalam mengeksekusi suatu perintah. Terdapat 2 hal yang harus dideklarasikan pada *script cron job*, yaitu waktu perulangan untuk mengeksekusi program dan letak direktori dari *file* perintah yang ingin dijalankan.

Terdapat 3 operator yang dapat digunakan untuk menentukan beberapa nilai dalam satu bagian waktu (menit, jam, tanggal dalam satu bulan, bulan, maupun hari), yaitu :

- Operator Bintang (\*) :

Operator ini menentukan semua nilai yang mungkin untuk suatu bagian waktu. Misalnya, operator bintang di bidang waktu jam akan sama dengan setiap jam atau operator bintang di bidang hari akan sama dengan setiap hari.

- Operator koma (,)

Operator ini menentukan daftar nilai, misalnya: "1,5,10,15,20, 25". Operator ini akan membuat suatu bagian waktu memiliki lebih dari satu nilai apabila dideklarasikan tanpa spasi.

- Operator hubung (-)

Operator ini menentukan rentang dari suatu nilai, misalnya: "5-15" hari, yang setara dengan mengetikkan "5,6,7,8,9, ..., 13,14,15" apabila kita menggunakan operator koma.

Dengan adanya fitur *Cron Job* pada Cpanel memudahkan pengguna dalam menuliskan *script* untuk melakukan pengaturan jadwal eksekusi suatu perintah.

### 2.2.8. Rapid Application Development (RAD)

Pengembangan aplikasi cepat atau yang lebih dikenal dengan sebutan RAD merupakan salah satu metode *agile software development*. RAD memiliki konsep yang hampir sama dengan metode *prototyping*. Keduanya memiliki tujuan yang sama, yaitu memperpendek waktu yang diperlukan dalam SDLC tradisional antara desain dan implementasi suatu sistem atau aplikasi [13]. Perbedaan dari kedua metode pengembangan ini adalah RAD lebih bersifat fleksibel, sehingga tahap *prototyping* dapat dilakukan secara berulang kali (iterasi). Hal ini memudahkan kedua belah pihak, baik pihak pengembang yang dimudahkan dalam proses pembuatan aplikasi apabila *requirement* detail tiap fitur bersifat fleksibel atau dapat berubah-ubah menyesuaikan dengan kebutuhan pihak perusahaan maupun dari pihak perusahaan yang dapat memantau perkembangan pengembangan aplikasi secara *real time*.

Pada dasarnya metode RAD memiliki beberapa jenis tahapan proses pengembangannya, namun secara general, semua jenis metode RAD minimal memiliki beberapa tahapan dibawah ini, yaitu :

#### 1. Requirements Planning

Pada tahap ini, pihak pengguna / perusahaan bertemu dengan pihak pengembang untuk membahas tujuan dan harapan pihak pengguna untuk proyek pengembangan aplikasi atau sistem serta masalah yang dihadapi oleh perusahaan saat ini . Selain itu, pada tahapan ini juga dibahas mengenai kebutuhan-kebutuhan apa saja yang diperlukan baik dari pihak pengembang maupun dari pihak perusahaan selama

proses pengembangan aplikasi berlanjut. Finalisasi persyaratan dengan persetujuan dari masing-masing pihak (perusahaan dan pengembang) menjadi sangat penting. Hal ini menunjukkan bahwa setiap orang memiliki kesempatan untuk mengevaluasi tujuan dan harapan untuk proyek pengembangan aplikasi serta mempertimbangkan segala resiko dari setiap keputusan yang diambil [14]. Dengan adanya hal ini diharapkan dapat meminimalisir miskomunikasi antar masing-masing pihak.

## 2. *RAD Design*

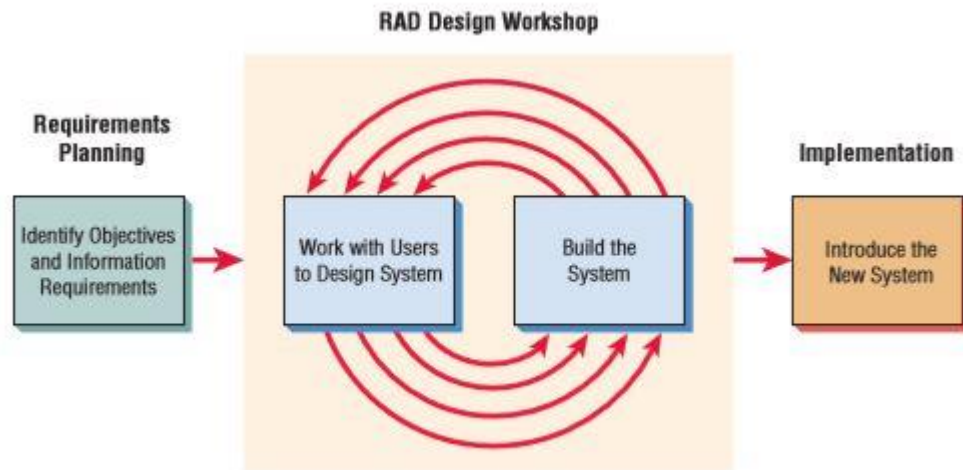
Tahap *RAD Design* merupakan inti dari metode pengembangan RAD di mana semua pihak, baik perusahaan maupun pengembang membangun desain pengguna melalui berbagai iterasi prototipe [13]. Selama tahap ini, perusahaan bekerja bahu-membahu dengan pengembang untuk memastikan kebutuhan mereka terpenuhi di setiap langkah dalam proses desain. Hal ini dilakukan agar pengembangan perangkat lunak dapat disesuaikan dengan kebutuhan, di mana pihak pengguna dapat menguji setiap prototipe produk, pada setiap langkah, untuk memastikan aplikasi tersebut dapat memenuhi harapan mereka.

Semua fase pengerjaan baik seperti penambahan fitur, penambahan parameter fungsi, perbaikan *bug* dikerjakan dalam proses berulang [14]. Pihak pengembang mendesain prototipe, pihak pengguna mengujinya, dan kemudian kedua belah pihak saling berkomunikasi tentang bagian apa yang berhasil dan bagian apa yang tidak. Metode ini memberi pihak pengembang kesempatan untuk mengubah model hingga mencapai desain yang memuaskan.

Tahap ini menjadi sangat penting karena pihak pengguna masih dapat memberi masukan selama proses berlangsung. Mereka dapat menyarankan perubahan - perubahan, atau bahkan ide-ide baru yang dapat menyelesaikan masalah mereka yang baru saja ditemukan.

## 3. *Implementation*

Tahapan ini merupakan tahapan terakhir di mana produk (aplikasi atau sistem) yang sudah jadi diluncurkan [13]. Hal ini sudah mencakup perubahan data, pengujian aplikasi atau sistem, dan pergantian ke sistem atau aplikasi yang baru, serta pelatihan penggunaan aplikasi dari pihak pengembang kepada pihak pengguna.



Gambar 2.3 Alur Rapid Application Development (RAD) [13]

### 2.2.9. Aplikasi Berbasis Website

Aplikasi berbasis *website* atau yang sering disebut *web apps* merupakan aplikasi yang mampu diakses melalui internet. *Web apps* banyak digunakan perusahaan-perusahaan untuk menunjang kebutuhan perusahaan. Berbagai macam fungsi dapat dikerjakan melalui *web apps* seperti mengirimkan *email*, membuat *invoice*, merekap data, dan banyak lainnya. Selain hal tersebut, *web apps* memiliki keunggulan dalam hal performa komputasi dan penyimpanan data karena tempat penyimpanan data pada *web apps* berada pada *cloud* (internet) [15].

### 2.2.10. Domain and Hosting

Suatu *website* harus memiliki domain dan hosting agar dapat diakses oleh pengguna internet lainnya. Keduanya memiliki peran vital masing-masing dalam pengembangan suatu *website*. Untuk lebih detailnya akan di paparkan dibawah ini.

Domain merupakan alamat *website* hasil terjemahan dari sebuah IP address yang dilakukan oleh DNS. Penerjemahan ini dilakukan agar dapat memudahkan pengguna internet dalam mengakses alamat IP dari suatu *website*. Ada banyak varian domain, namun tidak semuanya bebas digunakan. Terdapat beberapa domain yang memerlukan syarat agar pengguna dapat menggunakannya seperti .co.id atau .ac.id dan ada juga yang bersifat eksklusif seperti .desa atau .go.id (hanya untuk pemerintahan). Walaupun demikian masih banyak pilihan domain yang dapat digunakan oleh pengguna tanpa syarat yaitu seperti .com, .id, .xyz, .me, dan masih banyak lainnya.

Sedangkan Hosting atau *Web Hosting* adalah layanan yang memungkinkan suatu organisasi maupun individu untuk mengirim situs *web* atau halaman *web* mereka ke internet atau dengan kata lain hosting merupakan layanan yang mampu menyimpan data-data dari sebuah *website* agar dapat diakses melalui internet [16]. Situs web disimpan pada komputer khusus yang disebut server. Ketika pengguna internet ingin melihat suatu situs *web*, yang perlu mereka lakukan adalah mengetikkan alamat situs *web* atau domain dari *web* tersebut pada browser. Setelah itu komputer akan terhubung ke server *web* tersebut dan halaman *web* akan dikirimkan kepada mereka melalui browser.

Selain fungsi utama dari sebuah hosting adalah untuk penyimpanan data *website*, hosting juga memiliki beberapa fitur tambahan didalamnya sesuai dengan layanan penyedia hosting. Berikut adalah beberapa fitur yang dimiliki oleh sebagian besar penyedia layanan hosting :

- *FTP access*

Penggunaan FTP memungkinkan kita untuk mengunggah *file* dari komputer lokal ke server *web*. Sebagai contoh, apabila kita membangun satu situs *web* menggunakan *file* HTML pada laptop atau PC kita, maka kita dapat mentransfer file HTML tersebut dari komputer ke server *web* melalui FTP, yang memungkinkan situs *web* yang kita buat sebelumnya dapat diakses melalui internet.

- *Email Account*

Fitur akun *email* ini berfungsi untuk membuat email dengan nama domain yang terdaftar pada hosting yang digunakan. Sebagai contoh, apabila kita memiliki hosting dan mendaftarkan domain [www.websitekita.com](http://www.websitekita.com) , maka kita akan dapat membuat akun *email* dengan domain tersebut, seperti [user@websitekita.com](mailto:user@websitekita.com) atau [admin@websitekita.com](mailto:admin@websitekita.com). Hal ini sering digunakan sebagai email identitas dari suatu perusahaan maupun organisasi dalam mendaftar atau mengirimkan pekerjaan yang mengatasnamakan perusahaan atau organisasi tersebut.

- *Support Wordpress*

WordPress merupakan sistem manajemen konten (CMS). Dengan sistem CMS tersebut, pengguna sangat dimudahkan dalam membuat dan mengelola situs *web* mereka (karena bersifat *drag and drop*). WordPress memiliki lebih dari 25% situs *web* di internet. Sebagian besar penyedia hosting akan segera memberi tahu Anda jika paket mereka kompatibel dengan WordPress atau tidak. Persyaratan sederhana untuk

hosting situs *web* WordPress Anda meliputi: PHP versi 7 atau lebih tinggi; MySQL versi 5.6 atau lebih tinggi.

- *Scheduler*

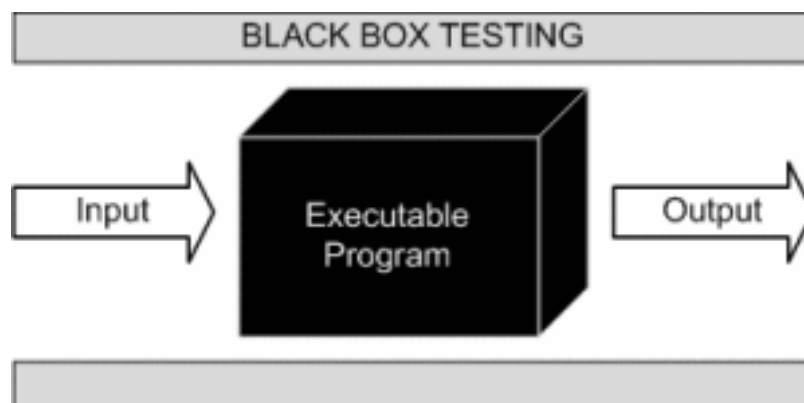
*Scheduler* berfungsi sebagai sistem penjadwalan pengeksekusian suatu perintah atau suatu *function* dari kode yang dibuat. Sistem penjadwalan ini bekerja berdasarkan 2 parameter yang harus diatur terlebih dahulu, yaitu waktu penjadwalan pengeksekusian program dan lokasi dari program yang akan dieksekusi.

- *SSL / TLS Manager*

Dengan *SSL / TLS Manager*, pengguna dapat *generate key* yang diperlukan dalam mengaktifkan protokol HTTPS terhadap *website* yang didaftarkan pada hostingan tersebut.

#### 2.2.11. Black Box Testing

*Black Box Testing* merupakan salah satu metode pengujian suatu fungsi aplikasi. Dalam pengujiannya, metode ini hanya menguji bagian luar dari suatu fungsi saja apakah alur yang dihasilkan sudah sesuai dengan yang diharapkan tanpa perlu mengetahui isi dari *logic* algoritma kodenya maupun struktur datanya [17]. Pada teknik ini, penguji hanya memasukkan *input* dari suatu fungsi, lalu mengamati hasil dan memastikan keluaran hasil prosesnya benar.



Gambar 2.4 Methode *Black Box Testing*

Terdapat beberapa teknik yang dapat digunakan dalam melakukan atau mengimplementasikan *black box testing* [18], yaitu :

- *Equivalence Partitioning*: Ini merupakan teknik desain uji perangkat lunak yang melibatkan pembagian nilai *input* ke dalam partisi yang valid dan tidak valid serta memilih nilai representatif yang akan digunakan sebagai data uji dari setiap partisi.
- *Boundary Value Analysis*: Ini merupakan teknik desain uji perangkat lunak yang menggunakan penentuan batas bawah maupun batas atas yang valid untuk nilai masukannya.
- *Cause-Effect Graphing*: Ini merupakan teknik desain uji perangkat lunak yang berfokus pada hasil *output* dari suatu proses berdasarkan kondisi *input* tertentu. Hasil dari pembuatan kasus uji sebab – akibat ini akan digambarkan dalam bentuk grafik maupun tabel.

Metode pengujian *black box* ini dipilih karena pihak perusahaan tidak terlalu mementingkan algoritme proses setiap fungsi atau fitur, melainkan hanya melihat hasil dari fungsi tersebut apakah sudah sesuai dengan ekspektasi perusahaan.



## BAB III

### METODE PENELITIAN

#### 3.1 Alat dan Bahan

##### 3.1.1. Alat

Alat yang dibutuhkan pada tugas akhir ini adalah sebagai berikut :

1. *Notebook* dengan spesifikasi minimum sistem operasi Windows 8, *processor* Intel Core i3 2330M CPU @ 2,2 GHz, memori 2GB DDR3, grafis AMD RADEON, *hardisk* 500GB. Pada tugas akhir ini digunakan Windows 10, Intel Core i3-4030U CPU@1,9GHz , Memori 4GB DDR 3, grafis AMD RADEON.
2. Sublime Text 3 sebagai *text editor* untuk menuliskan *script*.
3. XAMPP sebagai penyedia MySQL dan server lokal yang bernama *apache*
4. CodeIgniter sebagai *framework* aplikasi berbasis website.
5. *Library* PHPExcel sebagai perantara dalam melakukan penyimpanan data ke *database* dari *file excel* yang diunggah.
6. *Library* PHPMailer untuk konfigurasi pengiriman *email*.
7. CORELDRAW 2017 untuk melakukan *editing* gambar dan mendesain tampilan *website*.

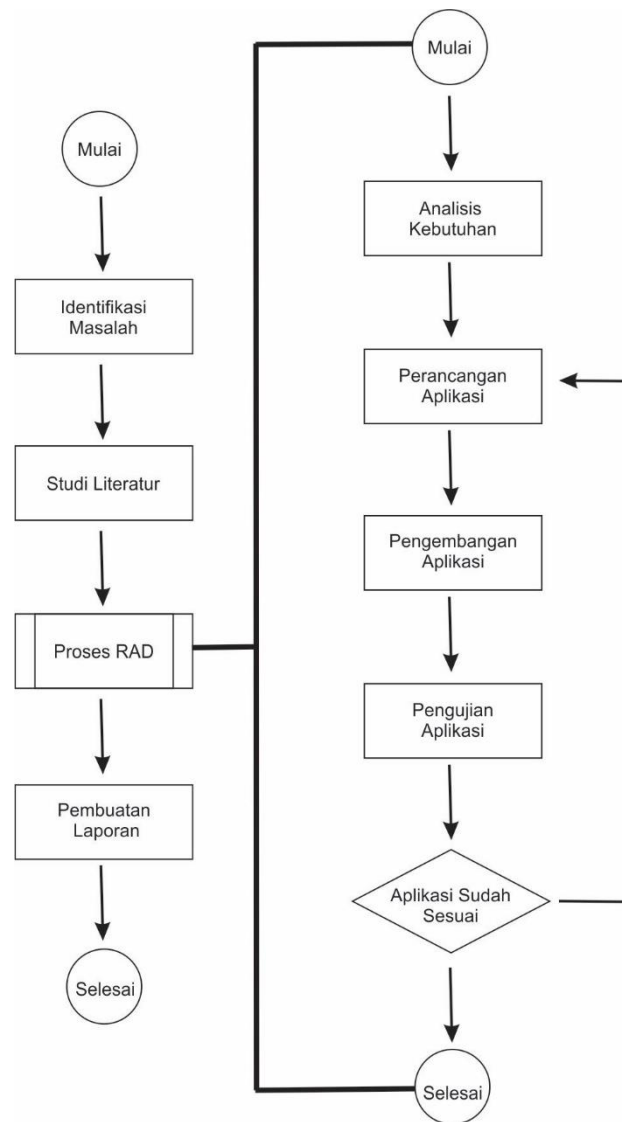
##### 3.1.2. Bahan

Bahan yang digunakan pada tugas akhir ini adalah :

1. Bahan yang berupa data atau informasi yang menjadi dataset tugas akhir. Dataset tugas akhir dapat berupa
  - Dataset *dummy* dari pihak PT. Hexindo Adiperkasa, Tbk.
  - Data hasil *interview* mengenai spesifikasi tiap fungsi pada aplikasi.
  - Dokumen panduan yang mengacu pada standar, hasil tugas akhir, atau artikel yang disitasi dan digunakan.

### 3.2 Alur Tugas akhir

Tugas Akhir ini dilakukan dengan tujuan untuk mengembangkan aplikasi pengingat tenggat waktu pembayaran hutang otomatis berbasis *website* oleh PT. Hexindo Adiperkasa Tbk. Aplikasi ini dikembangkan dengan metode RAD yang diilustrasikan pada Gambar 3.1.



Gambar 3.1 Alur Metode Penelitian

#### 3.2.1. Identifikasi Masalah

Tahap pertama yang dilakukan pada pengerjaan skripsi ini adalah identifikasi masalah. Pengindentifikasian masalah dilakukan dengan cara membahas bersama antara pihak perusahaan dengan pihak pengembang aplikasi mengenai permasalahan-permasalahan yang dihadapi oleh PT. Hexindo Adiperkasa, Tbk. . Pada tahap ini seluruh masalah dicatat dan dirangkum menjadi beberapa poin. Poin-poin

permasalahan tadi digunakan sebagai acuan dalam pengembangan fitur apa saja yang dibutuhkan oleh perusahaan.

### 3.2.2. Studi Literatur

Studi literatur dilakukan guna membandingkan, cara-cara apa saja yang sudah digunakan oleh pengembang lainnya untuk menyelesaikan masalah-masalah serupa yang dihadapi oleh PT. Hexindo Adiperkasa, Tbk. Hal ini akan menjadi pertimbangan pihak pengembang dalam mengembangkan fitur yang tepat guna menyelesaikan masalah yang dihadapi perusahaan secara optimal.

### 3.2.3. Analisis Kebutuhan

Pada tahap ini, pengembang mulai mengambil opsi - opsi yang akan digunakan untuk menyelesaikan masalah pada perusahaan. Semua fungsi yang akan dibuatpun sudah harus terpapar jelas pada tahap ini. Dengan adanya tahapan ini pihak pengembang sudah memiliki catatan dan gambaran mengenai hal-hal apa saja yang dibutuhkan dalam pengembangan aplikasi ini seperti nama domain, kebutuhan kapasitas memori hosting yang diperlukan, dan paket internet yang digunakan dalam pengembangan aplikasi. Semua hal tersebut dibahas bersama pihak perusahaan untuk dicarikan solusi bersama. Pada tahap ini dihasilkan daftar fitur apa saja yang dibutuhkan oleh pihak perusahaan yang dapat dilihat pada Tabel 3.1

Tabel 3.1 Daftar Fitur yang dibutuhkan pihak PT. Hexindo Adiperkasa Tbk.

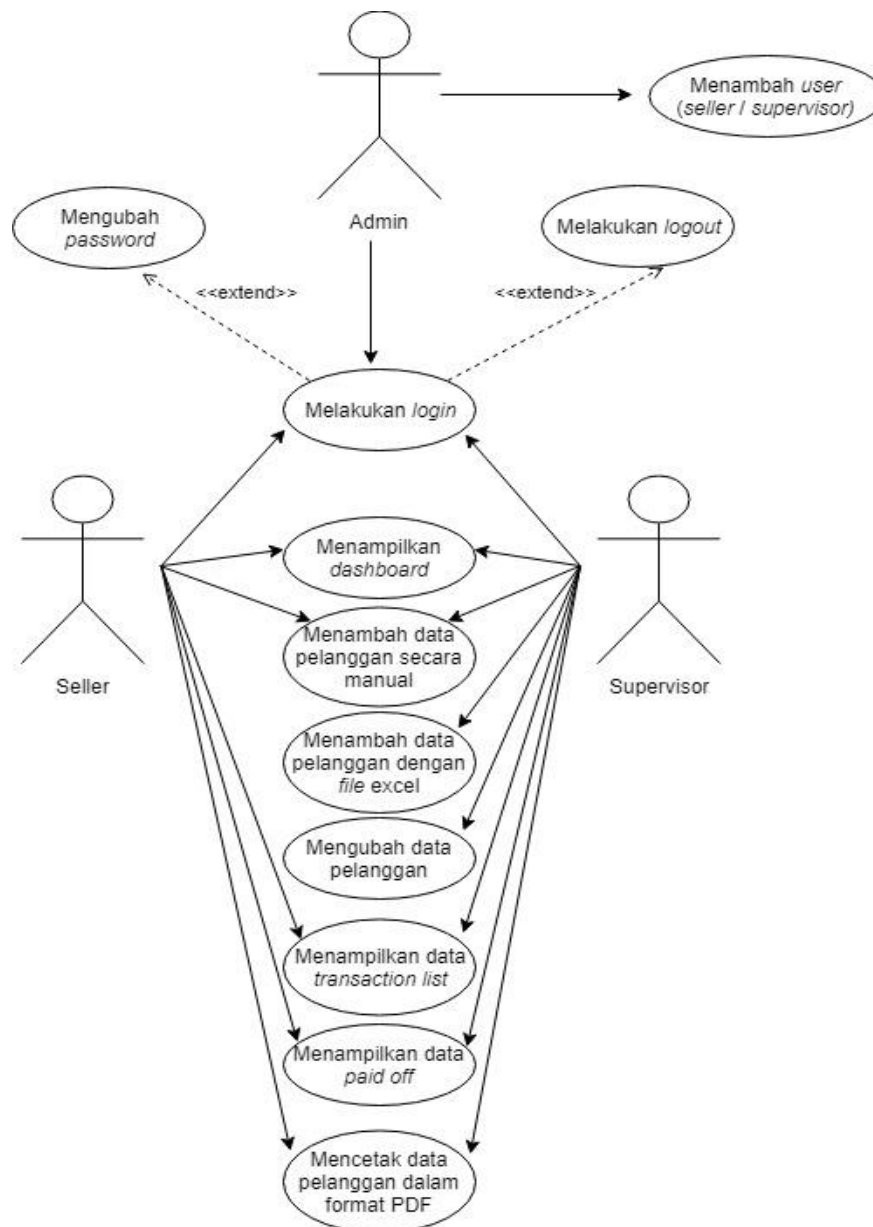
No.	Nama Fitur	Fungsi Fitur
1.	<i>Login Multilevel User</i>	Untuk melakukan <i>login</i> dan mengarahkan ke masing-masing halaman sesuai dengan <i>role</i> pengguna
2.	<i>Transaction List</i>	Untuk melihat daftar list pelanggan yang belum melunasi hutang ( <i>Supervisor</i> )
		Untuk melihat daftar list pelanggan yang belum melunasi hutang ( <i>Seller</i> )
3.	<i>Edit Data</i>	Untuk mengubah data pelanggan ( <i>Supervisor</i> )
4.	<i>Paid Off</i>	Untuk melihat daftar list pelanggan yang sudah melunasi hutang ( <i>Supervisor</i> )
		Untuk melihat daftar list pelanggan yang sudah melunasi hutang ( <i>Seller</i> )

No.	Nama Fitur	Fungsi Fitur
5.	<i>Input Data Manual</i>	Untuk memasukkan data pelanggan satu persatu ( <i>Supervisor</i> )
		Untuk memasukkan data pelanggan satu persatu ( <i>Seller</i> )
6.	<i>Input Data Excel</i>	Untuk memasukkan data pelanggan melalui <i>file excel</i> ( <i>Supervisor</i> )
7.	<i>Export to PDF</i>	Untuk merekap data dengan format keluaran berupa <i>file PDF</i> ( <i>Supervisor</i> )
		Untuk merekap data dengan format keluaran berupa <i>file PDF</i> ( <i>Seller</i> )
8.	<i>Dashboard</i>	Untuk melihat seluruh foto barang yang dijual oleh PT. Hexondo Adiperkasa Tbk. ( <i>Seller / Supervisor</i> ), tujuan <i>collection</i> , dan penyebab <i>overdue customer</i> .
9.	<i>Change Password</i>	Untuk meningkatkan keamanan setiap akun, sehingga baik <i>seller</i> maupun <i>supervisor</i> dapat mengubah <i>passwordnya</i> kapanpun sesuai yang ia inginkan
10.	Menambah akun baru	Untuk menambahkan akun <i>supervisor</i> atau <i>seller</i> baru ( <i>Admin</i> )
11.	<i>Reminder Otomatis</i>	Untuk mengirimkan peringatan waktu tenggat pembayaran secara otomatis melalui <i>email</i>

### 3.2.4. Perancangan Aplikasi

Hal pertama yang dilakukan pada tahap pengembangan aplikasi adalah membahas mengenai tampilan (*User Interface*) website. Pihak pengembang mengajukan beberapa tampilan UI yang dapat dipilih oleh pihak perusahaan sesuai dengan kebutuhannya. Setelah berdiskusi dan menentukan tampilan UI, pembahasan dilanjutkan kearah fungsionalitas aplikasi.

Pembahasan pertama adalah mengenai multilevel user atau ada berapa *role* yang diinginkan oleh perusahaan dan hal-hal apa saja yang dapat dilakukan maupun yang tidak dapat dilakukan oleh setiap *role*. Hal ini digambarkan melalui diagram *use case* yang dapat dilihat pada Gambar 3.2.

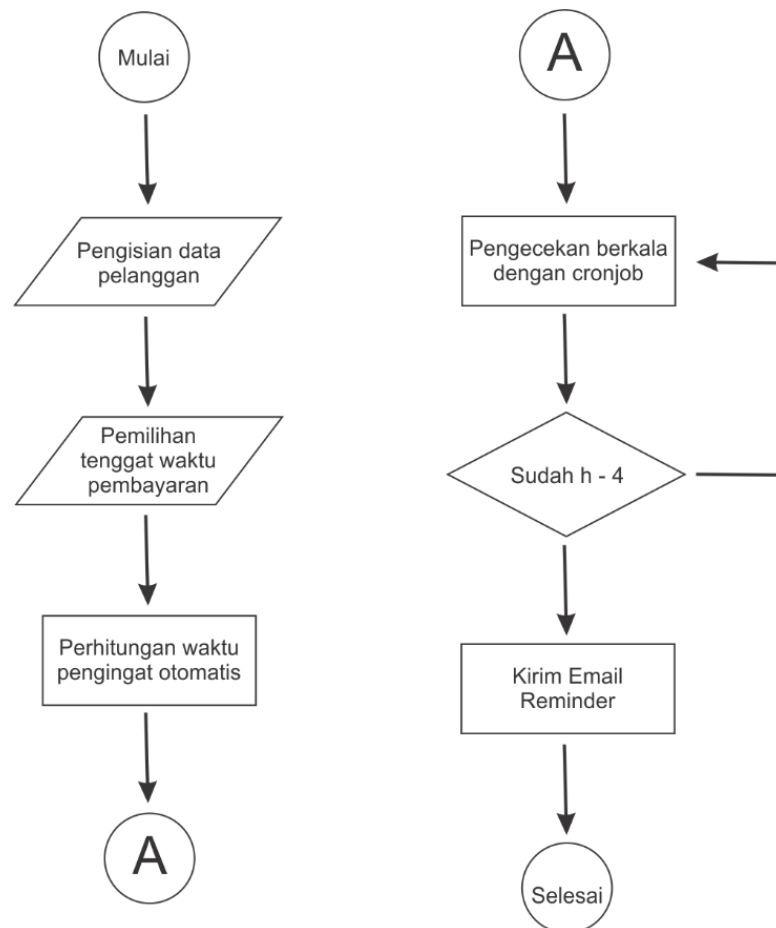


Gambar 3.2 Use Case Diagram seluruh Role

Berdasarkan Gambar 3.2, aplikasi ini digunakan oleh 3 role yaitu, *Admin*, *Supervisor*, dan *Seller*. *Admin* memiliki fungsi untuk menambahkan dan menghapus akun *seller* atau *supervisor*. Role ini hanya dipegang oleh satu orang pegawai. *Supervisor* memiliki peran untuk memantau seluruh data informasi pengguna secara keseluruhan maupun dari tiap-tiap *seller*. Informasi tersebut terdiri dari data lengkap pelanggan baik yang sudah maupun yang belum membayarkan hutang. *Supervisor* juga mampu menambahkan data pelanggan dengan menentukan *seller* mana yang mengurusnya serta mengubah data yang sudah ada. *Seller* dapat menambahkan data pelanggan yang melakukan transaksi melalui dirinya. *Seller* juga mampu melihat data pelanggan yang

bertransaksi melalui dirinya baik yang sudah melunasi hutang maupun yang belum. Selain itu, juga terdapat fitur login *multilevel user* di mana setiap *role* yang melakukan *login* akan diarahkan ke halaman masing-masing dari tiap *role*. Masing – masing akun dapat mengubah *password*-nya ketika sudah melakukan *login* terlebih dahulu. Pembahasan ini menjadi sangat penting karena fitur-fitur yang akan dikembangkan selanjutnya akan bergantung pada hak akses masing – masing *role* tersebut.

Setelah pembahasan mengenai *multilevel user* sudah jelas, pembahasan dilanjutkan dengan rancangan algoritme pengingat tenggat waktu pembayaran hutang otomatis yang akan dikirimkan kepada pelanggan. Alur yang digunakan sebagai pengingat dalam sistem ini ditunjukkan pada Gambar 3.3.



Gambar 3.3 Algoritme Pengingat Pembayaran Otomatis

- a) Pengisian data pelanggan dapat dilakukan dengan 2 cara, yaitu *seller* atau *supervisor* memasukkan data pelanggan satu persatu melalui fitur form yang

disediakan atau menggunakan *file* excel dengan *template* yang sudah disediakan pihak pengembang. Penginputan data melalui *file* excel akan memudahkan pihak perusahaan untuk menginputkan data dalam jumlah yang banyak dalam waktu yang singkat. Perbedaan kedua cara penambahan data ini terletak pada data yang perlu dimasukkan, apabila menggunakan fitur penambahan satu persatu, pihak perusahaan (baik *seller* maupun *supervisor*) tidak perlu menambahkan waktu tenggat pembayaran dan waktu *reminder*, karena sistem akan menghitung dan menentukan kedua waktu tersebut secara otomatis melalui tanggal penambahan data dan jenis tenggat waktu yang dipilih (akan dibahas di tahap selanjutnya), namun untuk penambahan data dengan menggunakan *file* excel pihak perusahaan harus sudah menentukan kedua waktu tersebut, karena sistem akan membaca tiap-tiap kolom dalam satu *file* excel tersebut dan memindahkannya ke *database* sistem. Oleh karena itu, maka dibutuhkan penyeragaman *file* excel yang akan diunggah melalui *template* yang sudah disesuaikan dengan *database* oleh pihak pengembang.

b) Jenis-jenis pemilihan tenggat waktu pembayaran ini sudah disesuaikan dengan standar perusahaan, di mana terdapat 7 jenis standar pemilihan tenggat waktu yaitu :

- CD = untuk tenggat waktu pembayaran keesokan harinya dari tanggal penambahan data (+ 1 hari)
- W1 = untuk tenggat waktu pembayaran 1 minggu dari tanggal penambahan data (+ 7 hari)
- W2 = untuk tenggat waktu pembayaran 2 minggu dari tanggal penambahan data (+ 14 hari)
- M1 = untuk tenggat waktu pembayaran sekitar 1 bulan dari tanggal penambahan data (+ 30 hari)
- MH = untuk tenggat waktu pembayaran sekitar 1 1/2 bulan dari tanggal penambahan data (+ 45 hari)
- M2 = untuk tenggat waktu pembayaran sekitar 2 bulan dari tanggal penambahan data (+ 60 hari)
- M3 = untuk tenggat waktu pembayaran sekitar 3 bulan dari tanggal penambahan data (+ 90 hari)

Dari ke 7 standar tersebutlah sistem akan secara otomatis menentukan kapan waktu tenggat pembayaran pelanggan, namun seperti yang sudah dijelaskan pada poin sebelumnya, bahwa perhitungan otomatis ini hanya bekerja ketika *seller* maupun *supervisor* menambahkan data melalui fitur *form input* data, bukan melalui *file excel*.

- c) Waktu *reminder* akan dihitung secara otomatis oleh sistem h-4 dari tenggat waktu pembayaran, namun sama halnya seperti poin sebelumnya, perhitungan waktu tenggat pembayaran otomatis ini hanya bekerja apabila *seller* maupun *supervisor* menambahkan data melalui fitur *form input* data, bukan melalui *file excel*.
- d) Setiap hari *cronjob* akan memanggil suatu fungsi untuk melakukan pengecekan apakah terdapat pelanggan yang sudah memasuki h-4 waktu tenggat pembayaran. Apabila terdapat pelanggan yang sudah memasuki h-4 waktu tenggat pembayaran, maka sistem akan secara otomatis mengirimkan *email reminder* kepada yang bersangkutan. Informasi yang dikirimkan berisi tentang jumlah hutang, waktu tenggat pembayaran, dan pemberitahuan bahwa hutang dapat dibayarkan dengan cara transfer ke rekening PT.Hexindo Adiperkasa, Tbk.

Pembahasan terakhir pada tahap ini adalah membahas alur dari tiap – tiap fitur secara mendetail. Dengan adanya rancangan dari *input – process – output* dari setiap fitur akan memudahkan pengerjaan (menentukan catatan-catatan apa saja yang diperlukan pada setiap fitur) dari pihak pengembang agar dapat memenuhi ekspektasi dari pihak perusahaan. Dari rancangan tersebut pula akan diketahui fitur – fitur yang mana sajakah yang memiliki hubungan dengan fitur lainnya.



### 3.2.5. Pengembangan Aplikasi

Pengembangan aplikasi dilakukan dengan mengacu pada daftar fitur yang sudah ditentukan melalui pembahasan pada tahap-tahap sebelumnya. Seluruh daftar fitur tersebut dapat dilihat pada Tabel 3.2

Tabel 3.2 Daftar Fitur yang akan dibuat

No.	Nama Fitur	Pembagian Fitur	Penjelasan fitur
1.	<i>Multilevel User</i>	Fitur <i>Login dan Multilevel User</i>	Fitur ini berfungsi sebagai pembatasan hak akses ( <i>privilege</i> ) dari masing-masing <i>role user</i> .
2.	<i>Transaction List</i>	Fitur <i>Transaction List (Supervisor)</i>	Fitur ini berfungsi sebagai pemantau data lengkap seluruh <i>customer</i> yang belum melakukan pembayaran.
		Fitur <i>Transaction List (Seller)</i>	Fitur ini berfungsi sebagai pemantau data lengkap <i>customer</i> (yang menggunakan perantara <i>seller</i> tersebut saja) yang belum melakukan pembayaran.
3.	<i>Edit Data</i>	Fitur <i>Edit Data (Supervisor)</i>	Fitur ini berfungsi untuk mengubah seluruh data <i>customer</i> .
4.	<i>Paid Off</i>	Fitur <i>Paid Off (Supervisor)</i>	Fitur ini berfungsi sebagai pemantau data lengkap seluruh <i>customer</i> yang sudah melakukan pembayaran.
		Fitur <i>Paid Off (Seller)</i>	Fitur ini berfungsi sebagai pemantau data lengkap <i>customer</i> (yang menggunakan perantara <i>seller</i> tersebut saja) yang sudah melakukan pembayaran.
5.	<i>Input Data Manual</i>	Fitur <i>Input Data Manual (Supervisor)</i>	Fitur ini berfungsi untuk menginputkan satu data <i>customer</i> (mampu memilih menggunakan perantara <i>seller</i> yang mana saja).
		Fitur <i>Input Data Manual (Seller)</i>	Fitur ini berfungsi untuk menginputkan satu data <i>customer</i> yang menggunakan perantara <i>seller</i> tersebut.
6.	<i>Input Data Excel</i>	Fitur <i>Input Data Excel (Supervisor)</i>	Fitur ini berfungsi untuk menginputkan banyak data <i>customer</i> (mampu memilih menggunakan perantara <i>seller</i> yang mana saja) sekaligus dalam bentuk <i>file excel</i> .

No.	Nama Fitur	Pembagian Fitur	Penjelasan fitur
7.	<i>Export to PDF</i>	Fitur <i>Export to PDF (Supervisor)</i>	Fitur ini berfungsi untuk merekap tabel pada fitur <i>Transaction List Supervisor</i> maupun <i>Paid Off Supervisor</i> dalam bentuk format <i>PDF</i> .
		Fitur <i>Export to PDF (Seller)</i>	Fitur ini berfungsi untuk merekap tabel pada fitur <i>Transaction List Seller</i> maupun <i>Paid Off Seller</i> dalam bentuk format <i>PDF</i> .
8.	<i>Dashboard</i>	Fitur <i>Dashboard</i>	Fitur ini berfungsi untuk menampilkan foto berbagai jenis barang yang dijual oleh PT. Hexindo Adiperkasa, Tbk., tujuan <i>collection</i> , dan penyebab <i>overdue customer</i> .
9.	<i>Change Password</i>	Fitur <i>Change Password</i>	Fitur ini berfungsi agar setiap akun <i>seller</i> maupun <i>supervisor</i> dapat mengubah <i>password</i> kapanpun yang ia mau. Hal ini akan meningkatkan keamanan dari tiap-tiap akun.
10.	Menambah akun baru	Fitur Menambah Akun Baru	Fitur berfungsi untuk menambahkan akun <i>supervisor</i> atau <i>seller</i> baru. Hal ini hanya dapat dilakukan oleh pengguna dengan <i>role</i> admin saja.
11.	<i>Reminder Otomatis</i>	Fitur <i>Reminder Otomatis</i>	Fitur ini berfungsi untuk mengingatkan <i>customer</i> mengenai tenggat waktu pelunasan pembayaran secara otomatis melalui <i>email</i> .

Pada tahap ini, fitur di kembangkan satu persatu oleh pihak pengembang. Hal ini dilakukan agar pihak pengembang dapat lebih fokus terhadap batasan – batasan dari setiap fitur dan menghasilkan fungsionalitas fitur secara maksimal.

### 3.2.6. Pengujian Aplikasi

Pada tahap ini, pengujian terhadap sistem aplikasi bertujuan untuk mengetahui apakah aplikasi yang dikembangkan ini sudah memenuhi segala kebutuhan dari pihak perusahaan sesuai dengan *requirement* yang sudah dibahas sebelumnya. Metode pengujian yang digunakan pada aplikasi ini adalah metode pengujian *black box*. Metode pengujian ini berfokus pada hasil keluaran dari setiap fungsi atau fitur tanpa memperdulikan proses algoritme yang berjalan di dalamnya. Oleh karena itu, hasil

keluaran yang diharapkan dari pihak perusahaan untuk tiap-tiap fungsi dipaparkan secara jelas melalui Tabel 3.3

Tabel 3.3 Daftar Kriteria Pengujian seluruh fitur

No.	Nama Fitur Diuji	Kriteria Evaluasi Hasil
1.	Fitur <i>Login dan Multilevel User</i>	Sistem dapat menampilkan halaman <i>Dashboard</i> yang sesuai dari masing – masing <i>role</i> berdasarkan <i>username</i> dan <i>password</i> yang dimasukkan. Apabila <i>username</i> atau <i>password</i> yang dimasukkan tidak terdapat pada database, maka aplikasi akan memberikan pemberitahuan bahwa <i>username</i> atau <i>password</i> yang dimasukkan salah.
2.	Fitur <i>Transaction List (Supervisor)</i>	Sistem mampu menampilkan seluruh daftar pengguna yang belum melakukan pembayaran hutang. Sistem juga dapat mengurutkan data, melakukan filter terhadap data pengguna berdasarkan nama <i>seller</i> maupun nama pengguna, mengubah data pengguna dari <i>Transaction List</i> ke <i>Paid Off</i> , serta menghapus data tersebut.
3.	Fitur <i>Transaction List (Seller)</i>	Sistem mampu menampilkan seluruh daftar pengguna yang belum melakukan pembayaran hutang (yang menggunakan perantara <i>seller</i> tersebut saja). Sistem juga dapat mengurutkan data dan melakukan filter terhadap data pengguna berdasarkan nama pengguna (yang menggunakan perantara <i>seller</i> tersebut saja).
4.	Fitur <i>Edit Data (Supervisor)</i>	Sistem mampu menampilkan data pelanggan, mengubah data tersebut, serta menyimpannya kembali ke <i>database</i> .
5.	Fitur <i>Paid Off (Supervisor)</i>	Sistem mampu menampilkan seluruh daftar pengguna yang sudah melakukan pembayaran hutang. Sistem juga dapat mengurutkan data, melakukan filter terhadap data pengguna berdasarkan nama <i>seller</i> maupun nama pengguna, serta menghapus data tersebut.
6.	Fitur <i>Paid Off (Seller)</i>	Sistem mampu menampilkan seluruh daftar pengguna yang sudah melakukan pembayaran hutang (yang menggunakan perantara <i>seller</i> tersebut saja). Sistem juga dapat mengurutkan

No.	Nama Fitur Diuji	Kriteria Evaluasi Hasil
		data dan melakukan filter terhadap data pengguna berdasarkan nama pengguna.
7.	Fitur <i>Input Data Manual (Supervisor)</i>	Sistem mampu memasukkan satu data pelanggan dan menyimpannya ke dalam <i>database</i> .
8.	Fitur <i>Input Data Manual (Seller)</i>	Sistem mampu memasukkan satu data pelanggan (yang menggunakan perantara <i>seller</i> tersebut saja). dan menyimpannya ke dalam <i>database</i> .
9.	Fitur <i>Input Data Excel (Supervisor)</i>	Sistem mampu memasukkan beberapa data pelanggan sekaligus dengan format <i>file excel</i> yang sudah ditentukan dan menyimpannya ke dalam <i>database</i>
10.	Fitur <i>Export to PDF (Supervisor)</i>	Sistem mampu mencetak daftar data pelanggan baik dalam tabel <i>Transaction List</i> maupun <i>Paid Off</i> pada <i>Dashboard Supervisor</i> dalam bentuk keluaran <i>PDF</i> .
11.	Fitur <i>Export to PDF (Seller)</i>	Sistem mampu mencetak daftar data pelanggan baik dalam tabel <i>Transaction List</i> maupun <i>Paid Off</i> pada <i>Dashboard Seller</i> dalam bentuk keluaran <i>PDF</i> .
12.	Fitur <i>Dashboard</i>	Sistem mampu menampilkan tujuan <i>collection</i> , dan penyebab <i>overdue customer</i> , serta foto berbagai jenis barang yang dijual oleh PT. Hexindo Adiperkasa, Tbk.
13.	Fitur <i>Change Password</i>	Sistem mampu mengubah <i>password</i> ketika terdapat akun <i>seller</i> maupun <i>supervisor</i> sudah melakukan <i>login</i> dan ingin mengubah password akunnya sendiri.
14.	Fitur Menambah Akun Baru	Sistem mampu memberikan akses kepada pengguna dengan <i>role</i> admin untuk menambahkan akun <i>supervisor</i> atau <i>seller</i> baru.
15.	Fitur <i>Reminder</i> Otomatis	Sistem mampu mengirimkan <i>reminder</i> berupa <i>email</i> yang berisikan tagihan, <i>invoice</i> , tanggal tenggat waktu pembayaran, dan informasi pengguna secara otomatis apabila terdapat pelanggan yang sudah memasuki h-4 waktu tenggat pembayaran.

Pada proses ini masukan dari pihak perusahaan akan terus diperhatikan seperti pada **Lampiran 1** dan **Lampiran 2** dan diuji kembali sampai hasil keluaran sudah sesuai dengan kriteria yang diinginkan oleh pihak PT. Hexindo Adiperkasa, Tbk.

## BAB IV

### HASIL DAN PEMBAHASAN

Bab ini akan memaparkan hasil pengembangan dan pengujian terhadap aplikasi yang dikembangkan pada tugas akhir ini. Hal yang akan dipaparkan antara lain mengenai hasil pembuatan dan implementasi basis data, tampilan antarmuka semua fitur beserta hasil pengujiannya dengan metode *black box*, serta detail dari pengingat tenggat waktu pembayaran hutang secara otomatis.

#### 4.1 Implementasi basis data

Dari seluruh daftar fitur-fitur yang akan dibuat berdasarkan Tabel 3.2, pihak pengembang membuat 2 tabel untuk memenuhi segala kebutuhan fitur-fitur tersebut, yaitu tabel *user* dan penjualan. Untuk lebih jelasnya mengenai isi *field* dari masing-masing tabel dapat dilihat pada Tabel 4.1 dan Tabel 4.2 .

Tabel 4.1 Tabel *user*

No.	Nama Field	Tipe Data	Ukuran Data	Penjelasan
1.	id	int	200	<i>Field</i> ini berfungsi sebagai <i>Primary Key</i> (PK) dari tiap akun
2.	<i>username</i>	varchar	200	<i>Field</i> ini berfungsi menjadi salah satu acuan/parameter <i>role</i> suatu akun ketika <i>login</i> ke aplikasi
3.	<i>password</i>	varchar	200	<i>Field</i> ini berfungsi menjadi salah satu acuan/parameter <i>role</i> suatu akun ketika <i>login</i> ke aplikasi. <i>Password</i> ini akan dapat diubah oleh tiap2 akun ( <i>seller/supervisor</i> ) untuk meningkatkan keamanan akun pengguna.
4.	<i>email</i>	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan <i>email</i> pengguna
5.	hp	varchar	100	<i>Field</i> ini berfungsi untuk menyimpan nomor hp pengguna
6.	level	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan <i>role</i> pengguna. Parameter inilah yang akan menjadi acuan <i>session</i> ketika pengguna melakukan <i>login</i> ke aplikasi

Tabel 4.1 ini secara keseluruhan akan menyimpan data-data dari setiap akun. Beberapa *field* dari tabel *user* akan disimpan pada *session* ketika *user* melakukan *login* ke aplikasi. Hanya *role admin* yang dapat melakukan penambahan data ke Tabel 4.1.

Tabel 4.2 Tabel penjualan

No.	Nama Field	Tipe Data	Ukuran Data	Penjelasan
1.	id	int	250	<i>Field</i> ini berfungsi sebagai <i>Primary Key</i> (PK) dari tiap data peminjaman
2.	nama_peminjam	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan data nama pelanggan
3.	email_peminjam	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan data <i>email</i> pelanggan
4.	hp_peminjam	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan nomor hp pengguna
5.	invoice_date	date	-	<i>Field</i> ini berfungsi untuk menyimpan tanggal pembuatan <i>invoice</i>
6.	invoice_number	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan nomor <i>invoice</i>
7.	TOP	varchar	100	<i>Field</i> ini berfungsi untuk menyimpan pilihan <i>Term of Payment</i> (TOP) tiap data peminjaman
8.	biaya	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan jumlah biaya hutang yang harus dibayarkan oleh pelanggan / peminjam
9.	reminder	date	-	<i>Field</i> ini berfungsi untuk menyimpan tanggal <i>reminder</i> (pengiriman melalui <i>email</i> ) atau h-4 tanggal tenggat waktu pembayaran
10.	deadline	date	-	<i>Field</i> ini berfungsi untuk menyimpan tanggal tenggat waktu pembayaran
11.	nama_seller	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan nama <i>seller</i> yang berkomunikasi dengan peminjam
12.	email_seller	varchar	200	<i>Field</i> ini berfungsi untuk menyimpan <i>email seller</i> yang berkomunikasi dengan pelanggan
13.	remark	varchar	500	<i>Field</i> ini berfungsi untuk menyimpan seluruh catatan ataupun keterangan tambahan mengenai data peminjaman

No.	Nama Field	Tipe Data	Ukuran Data	Penjelasan
14.	status	varchar	100	Field ini berfungsi untuk menyimpan status dari data peminjaman, apakah sudah dilunasi ( <i>paid off</i> ) atau belum.

Tabel 4.2 secara keseluruhan akan menyimpan data-data dari setiap peminjaman biaya. Tabel 4.2 yang dijadikan acuan dalam *filtering* data baik pada *seller* maupun *supervisor* dan acuan dalam pengiriman *email* pengingat tenggat waktu pembayaran secara otomatis kepada pelanggan. Hanya *supervisor* dan *seller* yang mampu memasukkan data peminjaman ke tabel ini sesuai dengan kewenangan hak akses masing-masing *role*.

## 4.2 Pengujian tiap fungsi dengan metode *black box*

Pada bagian ini akan dipaparkan mengenai hasil dari pengujian masing-masing fitur berdasarkan Tabel 3.3 dengan metode *black box*. Beberapa diantara hasil pengujian tersebut akan ditampilkan dalam bentuk tampilan antarmuka serta beberapa penjelasan mengenai *back end script code*-nya.

### 4.2.1. Fitur login dan multilevel user

Pengujian pada fitur ini dilakukan dengan 4 skenario, dimana pengembang akan memasukkan *username* dan *password* untuk akun admin, *supervisor*, *seller*, dan salah satu atau kedua dari *username* atau *password* salah. Logika yang digunakan pada *script code back end* yang digunakan untuk *login* dan *multilevel user* ini dapat dilihat pada **Lampiran 3**.

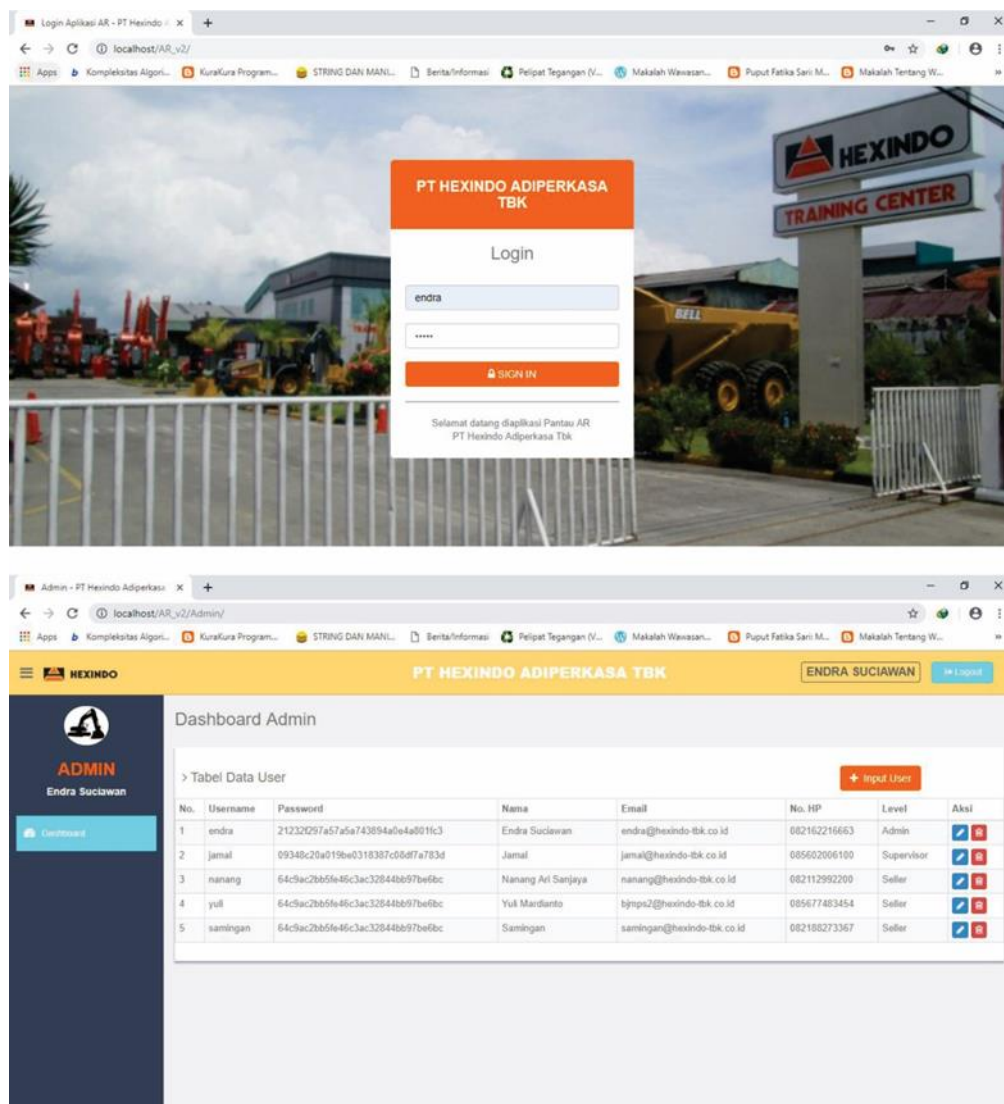
Secara singkat cara kerja logika *login multilevel user* ini adalah, sistem akan mencocokkan apakah *username* dan *password* yang dimasukkan telah terdaftar pada *database*. Apabila sudah terdaftar, sistem akan mencari *level* dari akun tersebut dan diteruskan ke tampilan *dashboard* yang sesuai dengan *role* yang dimasukkan dengan membawa beberapa variabel ke dalam *session*-nya seperti nama, *email*, nomor hp, level, serta id dari akun tersebut. Variabel-variabel inilah yang akan menjadi parameter ketika pengguna melakukan suatu aktivitas setelah *login*. Apabila *username* dan *password* yang dimasukkan tidak terdaftar pada *database*, sistem akan memberikan informasi pada halaman *login* bahwa *username* atau *password* yang dimasukkan oleh pengguna salah.



Dari pengujian ini akan terlihat cara kerja *multilevel user* sehingga halaman akan dilemparka sesuai dengan *role* yang dimasukkan serta menguji apakah sistem dapat merespon apabila pengguna tidak benar dalam memasukkan salah satu dari *username* dan *password* atau keduanya.

a) Menggunakan *username* dan *password* admin

Pada percobaan ini pengembang akan memasukkan *username* dan *password* dari pengguna dengan *role* admin. Hasil dari percobaan ini dapat dilihat pada Gambar 4.1.



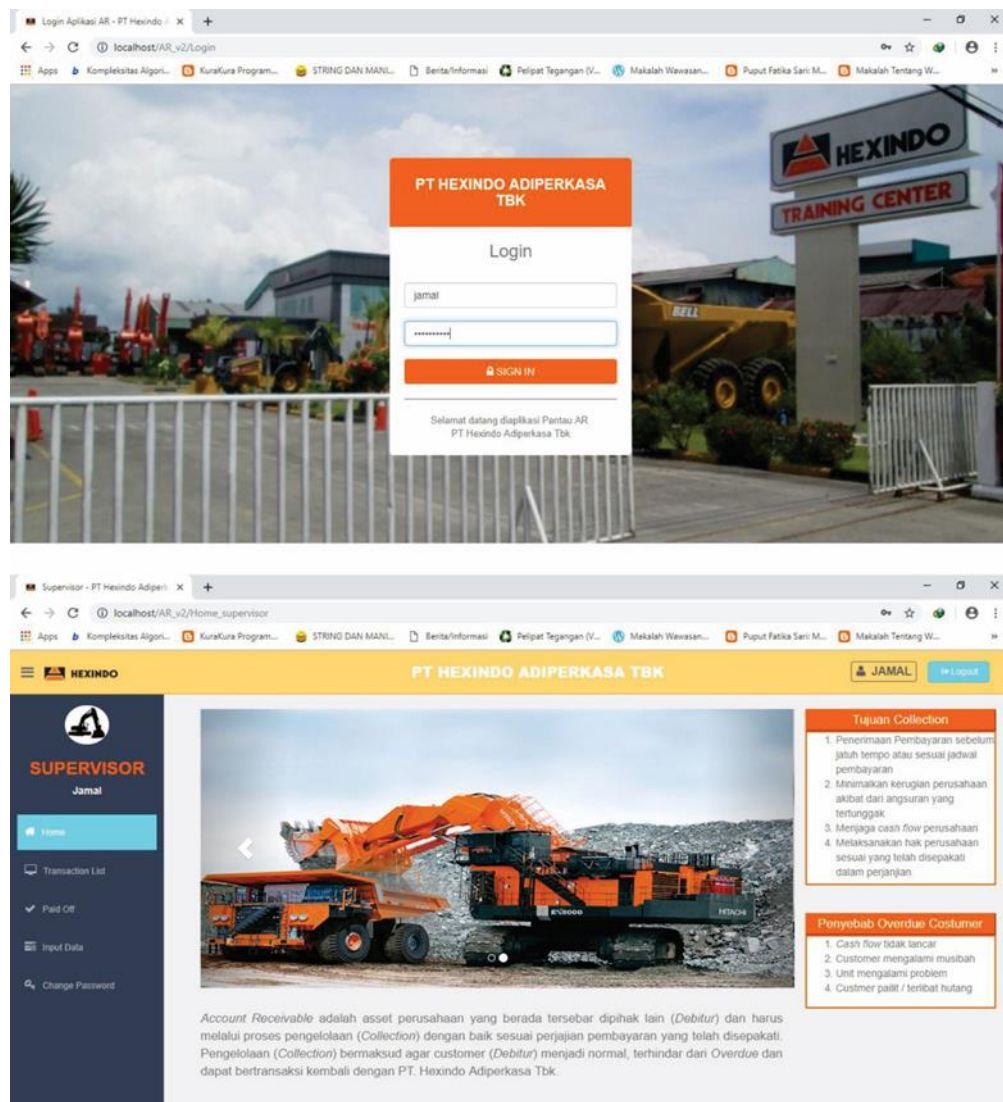
Gambar 4.1 Hasil uji coba *login* dengan menggunakan *username* dan *password* admin



Dari hasil uji login dengan *username* dan *password* admin, didapatkan hasil sesuai yang diharapkan di mana pengguna dengan *role* admin akan langsung disajikan tampilan *dashboard* admin.

b) Menggunakan *username* dan *password supervisor*

Pada percobaan ini pengembang akan memasukkan *username* dan *password* dari salah satu pengguna dengan *role supervisor*. Hasil dari percobaan ini dapat dilihat pada Gambar 4.2.

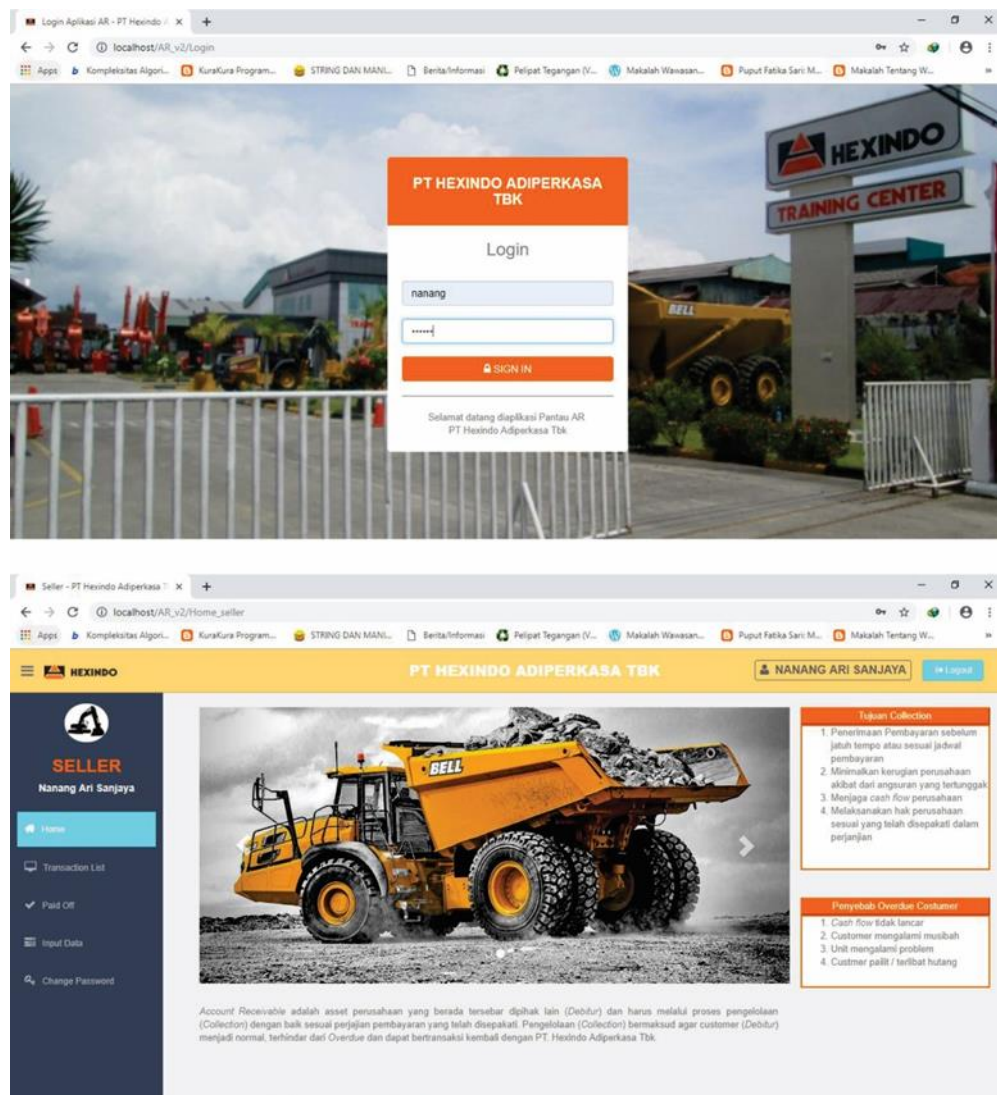


Gambar 4.2 Hasil uji coba *login* dengan menggunakan *username* dan *password supervisor*

Dari hasil uji login dengan *username* dan *password supervisor*, didapatkan hasil sesuai yang diharapkan di mana pengguna dengan *role supervisor* akan langsung disajikan tampilan *dashboard supervisor*.

c) Menggunakan *username* dan *password seller*

Pada percobaan ini pengembang akan memasukkan *username* dan *password* dari salah satu pengguna dengan *role seller*. Hasil dari percobaan ini dapat dilihat pada Gambar 4.3.

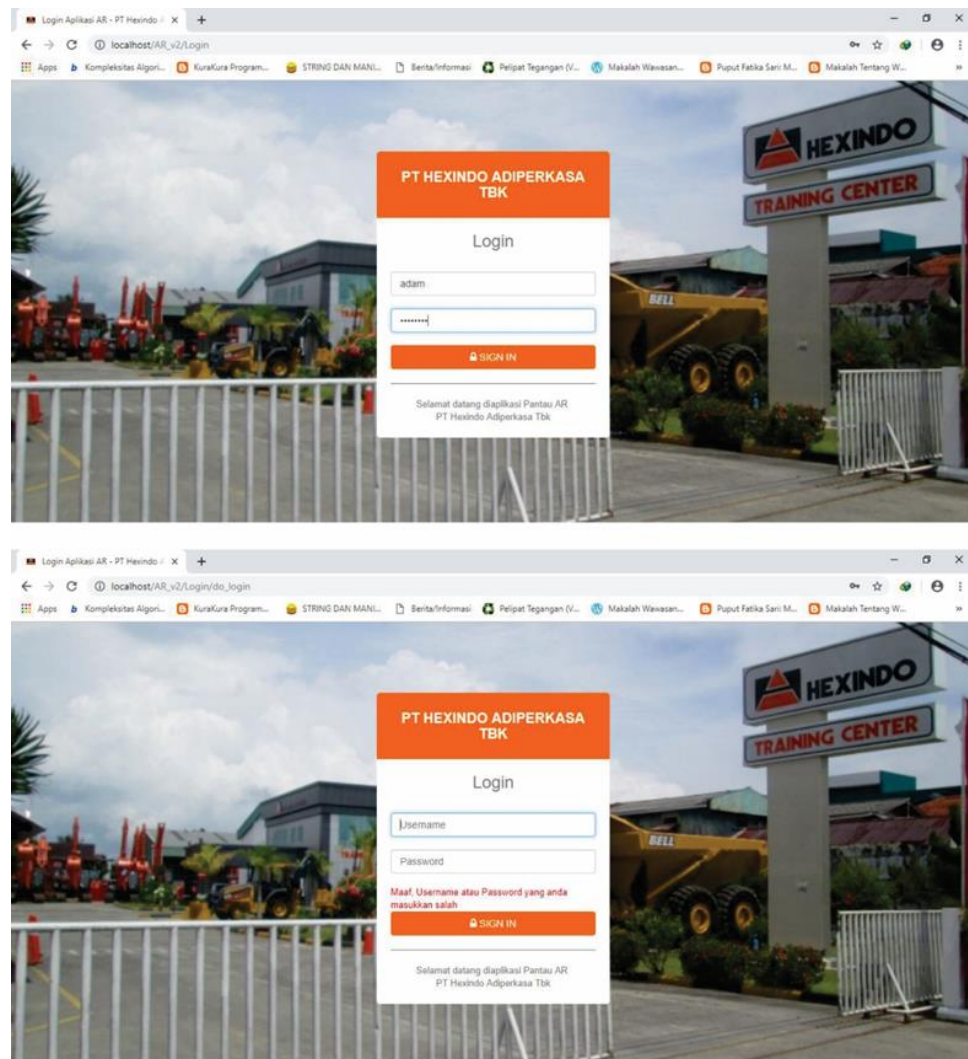


Gambar 4.3 Hasil uji coba *login* dengan menggunakan *username* dan *password seller*

Dari hasil uji login dengan *username* dan *password seller*, didapatkan hasil sesuai yang diharapkan di mana pengguna dengan *role seller* akan langsung disajikan tampilan *dashboard seller*.

d) Memasukkan *username* atau *password* yang salah

Pada percobaan ini pengembang akan memasukkan *username* atau *password* yang salah atau yang tidak terdaftar pada *database*. Hasil dari percobaan ini dapat dilihat pada Gambar 4.4.

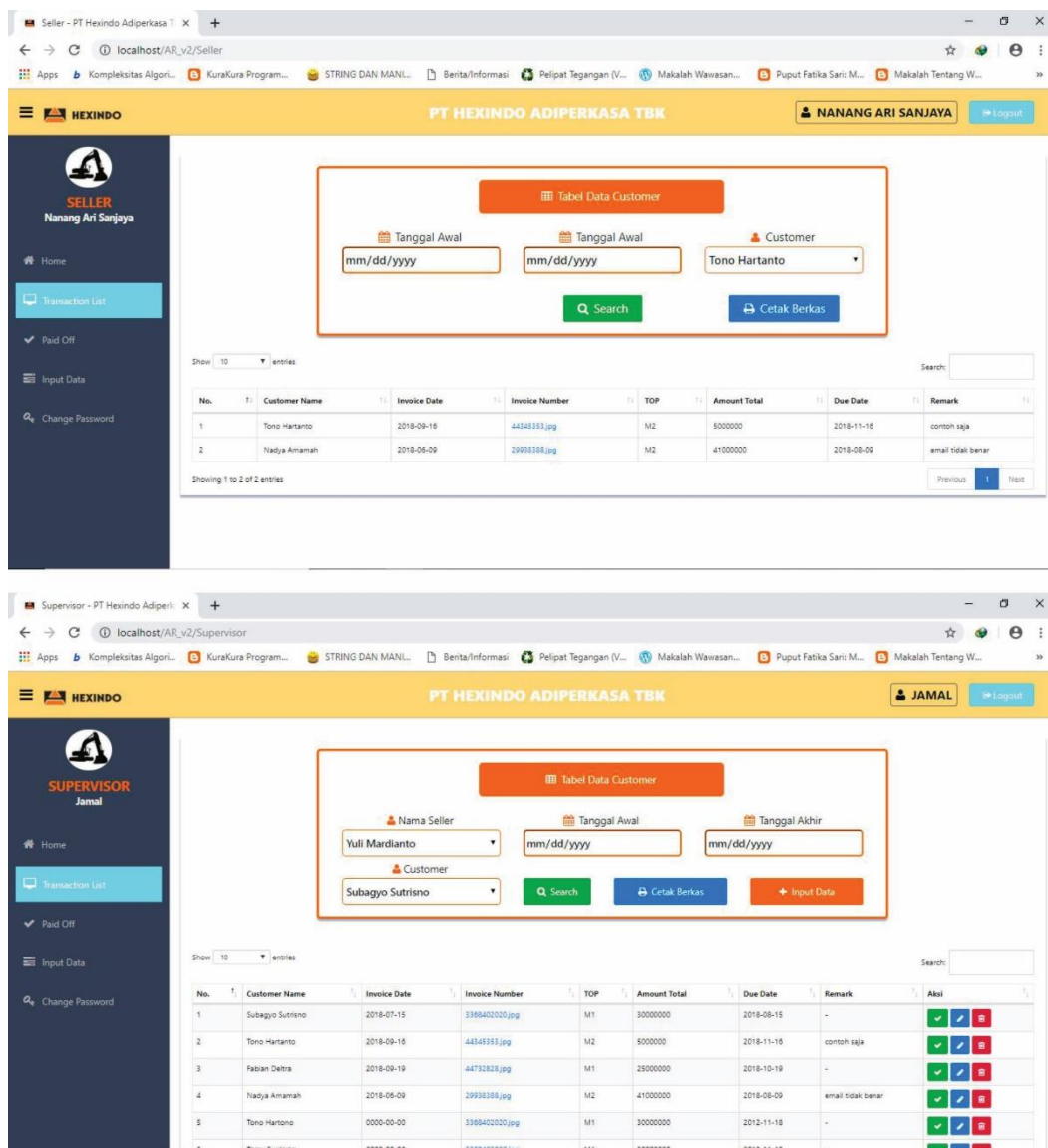


Gambar 4.4 Hasil uji coba login dengan menggunakan username atau password yang salah

Dari hasil uji login dengan *username* atau *password* yang salah, didapatkan hasil sesuai yang diharapkan di mana sistem akan memberikan pemberitahuan bahwa *username* atau *password* yang dimasukkan salah.

#### 4.2.2. Fitur Transaction List (Supervisor dan Seller)

Fitur ini benrbentuk suatu halaman yang berisikan daftar pelanggan yang belum melunasi hutang pembayaran terhadap perusahaan. Pengujian terhadap fitur ini dilakukan dengan cara mencocokkan apakah data yang ditampilkan pada sisi *supervisor* maupun *seller* merupakan data pelanggan yang belum melunasi hutangnya. Selain hal tersebut, sistem juga mampu mengurutkan data serta menyaring data berdasarkan *seller* maupun pelanggan untuk *role supervisor* dan menyaring pelanggan yang melalui dirinya untuk *role seller*.



Gambar 4.5 Tampilan transaction list seller (atas) dan supervisor (bawah)

Berdasarkan Gambar 4.5 dapat dilihat bahwa hasil yang ditampilkan pada fitur *transaction list* baik *seller* maupun *supervisor* sesuai yang diharapkan yaitu menampilkan data pelanggan yang belum melunasi hutang pembayaran. Data-data



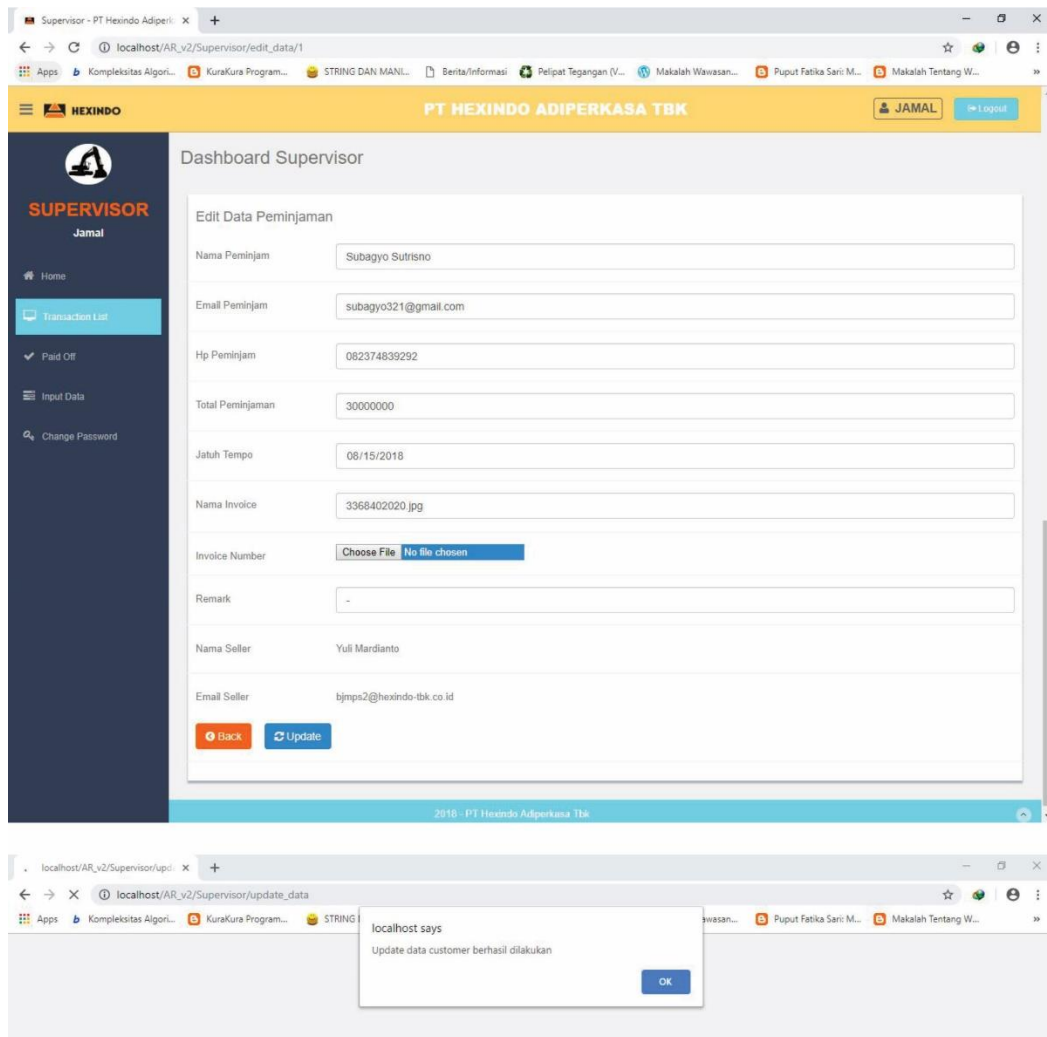
yang ditampilkan meliputi nama pelanggan, *invoice date*, *invoice number*, TOP, total hutang, waktu tenggat pembayaran, dan catatan terhadap pelanggan. Pada fitur ini juga baik *seller* maupun *supervisor* dapat mengurutkan data dengan cara mengklik tanda panah pada tiap kolom tabel dan melakukan filter data dengan parameter nama pelanggan dan rentan waktu yang diukur berdasarkan tengat waktu pembayaran pelanggan tersebut.

Terdapat 2 perbedaan dari tampilan *transaction list* akun *seller* dan *supervisor*. Perbedaan pertama dapat dilihat pada pada akun *seller* (atas), bahwa hasil yang ditampilkan merupakan data pelanggan yang bertransaksi melalui dirinya saja, sedangkan pada akun *supervisor* dapat melihat seluruh data pelanggan dari siapapun *seller*-nya. Hal ini ditunjukkan dengan adanya filter nama *seller* pada bagian atas akun *supervisor* yang berfungsi untuk melihat berapa banyak pelanggan yang melakukan transaksi melalui *seller* tertentu.

Perbedaan kedua adalah pada bagian tabel data yang berada dibawah. Pada akun *supervisor* terdapat kolom aksi yang terdri dari 3 tombol yaitu tombol *done* (untuk memindahkan data dari *transaction list* / hutang belum lunas ke *paid off* / hutang sudah lunas), tombol *edit* (untuk mengubah data pelanggan), dan tombol *delete* (untuk menghapus data pelanggan), sedangkan pada akun *seller* tidak ada kolom tersebut. Hal ini yang menjadi perbedaan hak akses data pelanggan dari kedua *role*.

#### **4.2.3. Fitur Edit Data**

Pengujian pada fitur ini dilakukan dengan cara *login* menggunakan salah satu akun *supervisor* lalu mengubah salah satu data pelanggan. Tampilan pada pengujian ini dapat dilihat pada Gambar 4.6.



Gambar 4.6 Tampilan fitur *edit* data pada akun *supervisor*

Pada Gambar 4.6 dapat dilihat bahwa hasil pengujian berhasil dilakukan dan sesuai dengan yang diharapkan, dimana *supervisor* mampu mengubah data dari pelanggan. Tampilan *form edit* data akan muncul ketika *supervisor* menekan tombol *edit* pada kolom aksi di *transaction list*. Lalu setelah *supervisor* selesai mengubah data, yang perlu dilakukan hanyalah menekan tombol *update*. Apabila data berhasil diubah, maka akan muncul *pop up* yang bertuliskan “Update data customer berhasil dilakukan”, namun apabila gagal, maka akan muncul *pop up* yang bertuliskan “Data gagal di update”. Sesuai dengan yang sudah dijelaskan sebelumnya bahwa fitur *edit* data ini hanya dapat dilakukan oleh akun *supervisor*, sehingga akan menjamin keamanan data pelanggan.

#### 4.2.4. Fitur Paid Off (Supervisor dan Seller)

Fitur ini berbentuk suatu halaman yang berisikan daftar pelanggan yang sudah melunasi hutang pembayaran terhadap perusahaan. Pengujian terhadap fitur ini dilakukan dengan cara mencocokkan apakah data yang ditampilkan pada sisi *supervisor* maupun *seller* merupakan data pelanggan yang sudah melunasi hutangnya. Selain hal tersebut, sistem juga mampu mengurutkan data serta menyaring data berdasarkan *seller* maupun pelanggan untuk *role supervisor* dan menyaring pelanggan yang melalui dirinya untuk *role seller*.

The top screenshot shows the Seller interface for Nanang Ari Sanjaya. It features a sidebar with 'Paid Off' selected and a main area with a 'Tabel Data Customer' form. The form includes input fields for 'Tanggal Awal' (mm/dd/yyyy), 'Tanggal Akhir' (mm/dd/yyyy), and a 'Customer' dropdown menu set to 'Semua'. Below the form is a table with columns: No., Customer Name, Invoice Date, Invoice Number, TOP, Amount Total, Due Date, and Remark. The table is currently empty, showing 'No data available in table'.

The bottom screenshot shows the Supervisor interface for Jamal. It features a sidebar with 'Paid Off' selected and a main area with a 'Tabel Data Customer' form. The form includes input fields for 'Nama Seller' (Yuli Mardianto), 'Tanggal Awal' (mm/dd/yyyy), 'Tanggal Akhir' (mm/dd/yyyy), and a 'Customer' dropdown menu set to 'Annisa Khansa'. Below the form is a table with columns: No., Customer Name, Invoice Date, Invoice Number, TOP, Amount Total, Due Date, Remark, and Aksi. The table contains one entry:

No.	Customer Name	Invoice Date	Invoice Number	TOP	Amount Total	Due Date	Remark	Aksi
1	Annisa Khansa	2018-06-19	17281828.jpg	M1	2700000	2018-07-19		<a href="#">Edit</a> <a href="#">Delete</a>

Gambar 4.7 Tampilan *paid off seller* (atas) dan *supervisor* (bawah)

Berdasarkan Gambar 4.7 dapat dilihat bahwa hasil yang ditampilkan pada fitur *paid off* baik *seller* maupun *supervisor* sesuai yang diharapkan yaitu menampilkan data

pelanggan yang sudah melunasi hutang pembayaran. Kolom yang ditampilkan pada fitur ini merupakan kolom yang sama dengan yang ditampilkan pada *transaction list* yaitu meliputi nama pelanggan, *invoice date*, *invoice number*, TOP, total hutang, waktu tenggat pembayaran, dan catatan terhadap pelanggan. Pada fitur ini juga baik *seller* maupun *supervisor* dapat mengurutkan data dengan cara mengklik tanda panah pada tiap kolom tabel dan melakukan filter data dengan parameter nama pelanggan dan rentan waktu yang diukur berdasarkan tengat waktu pembayaran pelanggan tersebut.

Pada fitur ini juga terdapat 2 perbedaan dari tampilan *transaction list* akun *seller* dan *supervisor*. Perbedaan pertama dapat dilihat pada pada akun *seller* (atas), bahwa hasil yang ditampilkan merupakan data pelanggan yang bertransaksi melalui dirinya saja, sedangkan pada akun *supervisor* dapat melihat seluruh data pelanggan dari siapapun *seller*-nya.

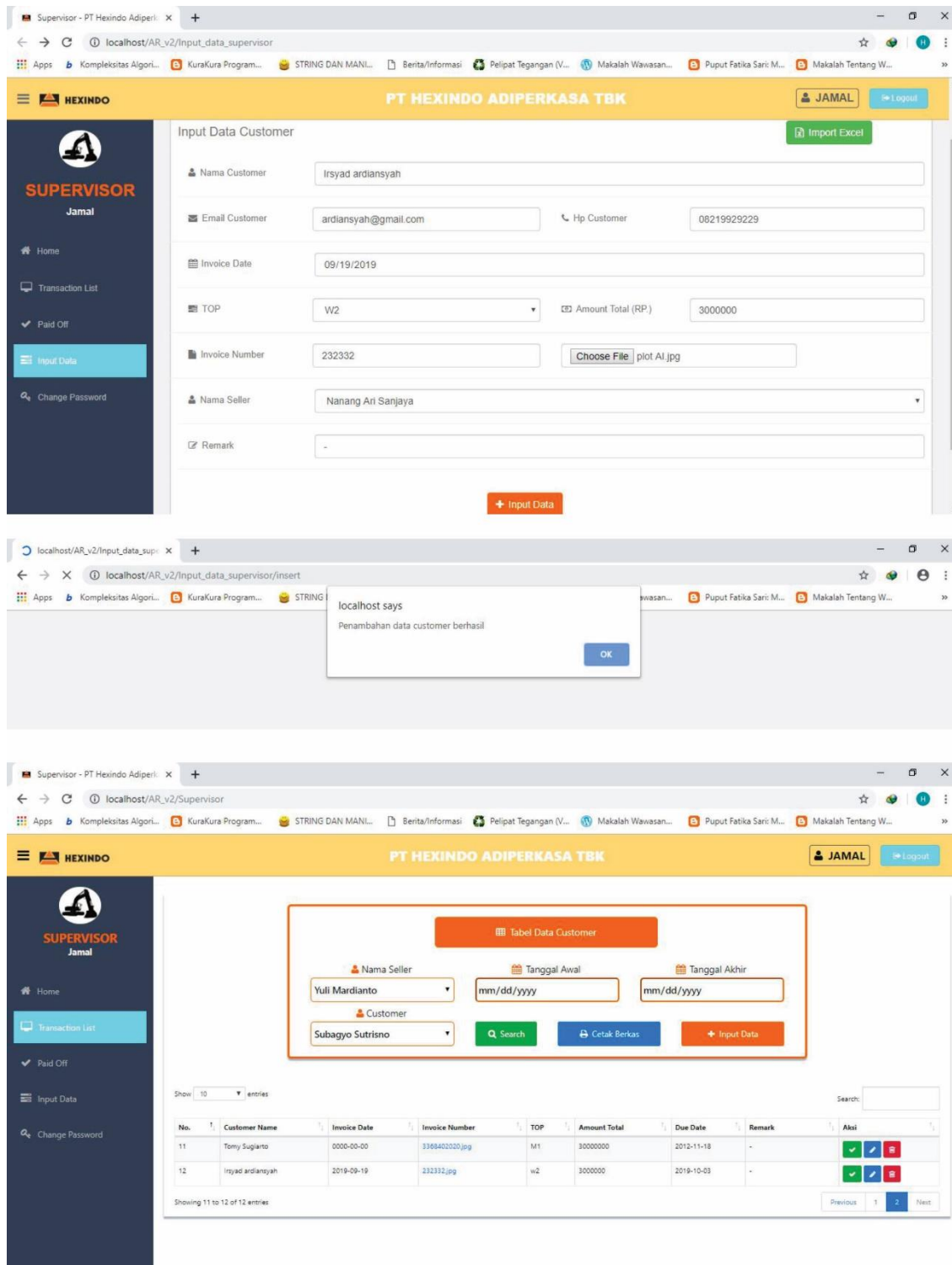
Perbedaan kedua adalah pada bagian tabel data yang berada dibawah. Pada akun *supervisor* terdapat kolom aksi yang terdri dari 2 tombol yaitu tombol *edit* (untuk mengubah data pelanggan) dan tombol *delete* (untuk menghapus data pelanggan), sedangkan pada akun *seller* tidak ada kolom tersebut.

Secara keseluruhan tampilan pada fitur ini mirip dengan fitur *transaction list*, yang membedakan kedua fitur ini adalah status pembayaran hutang dari pelanggan tersebut saja.

#### **4.2.5. Fitur Input Data Manual (Supervisor dan Seller)**

Fitur ini benrbentuk suatu halaman yang berisikan *form input* data. Pengujian terhadap fitur ini dilakukan dengan cara menambahkan data pelanggan secara manual atau satu persatu ke *database*. Data-data yang perlu diisi meliputi nama pelanggan, *email* pelanggan, nomor hp pelanggan, *invoice date*, TOP, total hutang, *invoice number* (nama + foto *invoice*), catatan untuk pelanggan, dan nama *seller* khusus untuk akun *supervisor* (apabila *login* dengan akun *seller*, nama *seller* tersebut sudah ditampilkan secara otomatis oleh sistem).





Gambar 4.8 Tampilan penginputan data secara manual

Pada Gambar 4.8 dapat dilihat secara sekilas bahwa bentuk *form* dari kedua *role* (*seller* atau *supervisor*) memiliki isian yang hamper sama. Terdapat 2 perbedaan pada *form input* manual pada kedua *role* tersebut. Perbedaan pertama adalah pada *role supervisor* terdapat tombol 'Import Excel', di mana tombol tersebut dapat berfungsi

untuk melakukan penambahan data secara masal dengan format *file* excel yang sudah ditentukan (fitur ini akan dibahas setelah sub bab ini), sedangkan pada *role seller* tidak ada. Perbedaan yang kedua adalah pada *role supervisor*, seorang *supervisor* mampu memilih *seller* siapa saja yang hendak melakukan proses transaksi dengan pelanggan tersebut, sedangkan pada *role seller* tidak dapat melakukan hal tersebut, karena *nama seller* sudah akan tertulis secara otomatis ketika *seller* tersebut ingin menambahkan data pelanggan.

Pengujian yang dilakukan terhadap fitur ini berhasil yang ditandai dengan munculnya *pop up* bertuliskan “Penambahan data customer berhasil” sesuai pada Gambar 4.8 dan data yang baru ditambahkan akan langsung tersimpan dan ditampilkan pada tabel *transaction list*. *Pop up* tersebut akan muncul dengan tulisan yang sama ketika baik *seller* maupun *supervisor* berhasil melakukan penambahan data secara manual.

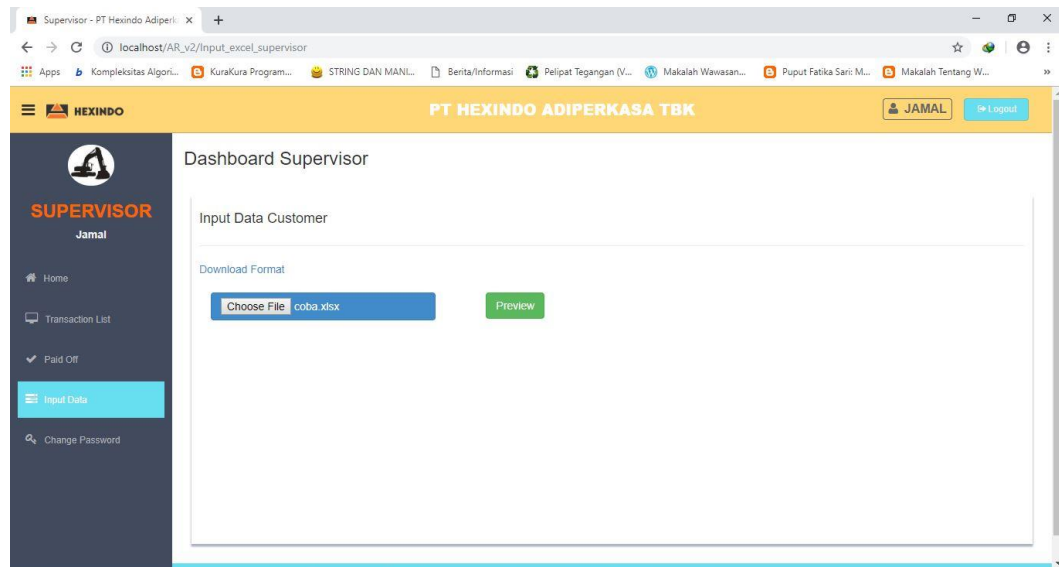
#### 4.2.6. Fitur Input Data Excel

Fitur ini memerlukan *library* tambahan yang bernama PHPExcel yang diletakkan di *third\_party* folder code igniter. Fungsi dari *library* ini adalah untuk menghubungkan antara *file* excel yang telah dirancang agar dapat terbaca dan disimpan ke *database*. Tahap selanjutnya adalah membuat kerangka format *file* excel. Perlu penyesuaian format tipe untuk setiap kolom *file* excel terhadap format kolom pada tabel yang akan dituju, dalam kasus ini adalah tabel penjualan yang akan dituju. Contoh dari format dari kerangka *file* excel yang sudah diisi 2 data dapat dilihat pada Gambar 4.9.

A	B	C	D	E	F	G	H	I	J	K	L	M
nama_peminjam	email_peminjam	hp_peminjam	invoice_date	invoice_number	TOP	biaya	reminder	deadline	nama_seller	email_seller	remark	status
Aliudin Raharjo	<a href="mailto:aliudin@gmail.com">aliudin@gmail.com</a>	085745345567	2019-07-16	3368020.jpg	M1	30000000	2019-08-12	2019-08-16	Yulii Mardianto	<a href="mailto:bjmps2@hexindo-tbk.co.id">bjmps2@hexindo-tbk.co.id</a>	-	Pending
Bagus Suwaryo	<a href="mailto:bagus@gmail.com">bagus@gmail.com</a>	082114958834	2019-06-20	3368120.jpg	M2	25000000	2019-08-16	2019-06-20	Nanag Ari Sanjaya	<a href="mailto:nanang@hexindo-tbk.co.id">nanang@hexindo-tbk.co.id</a>	-	Pending

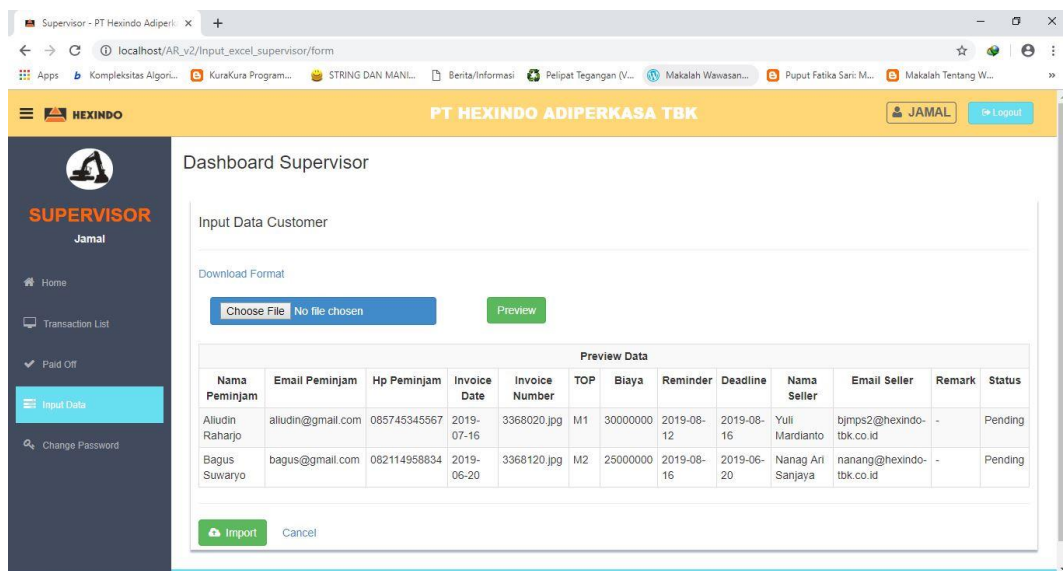
Gambar 4.9 Contoh penginputan data dengan *file excel*

Setelah mengisi data dengan format excel yang telah ditentukan, *supervisor* hanya perlu menekan tombol “Import Excel” pada halaman Input Data dan mengunggah *file* excel tersebut. Tahap ini dapat dilihat pada Gambar 4.10.



Gambar 4.10 Tampilan *Upload file excel*

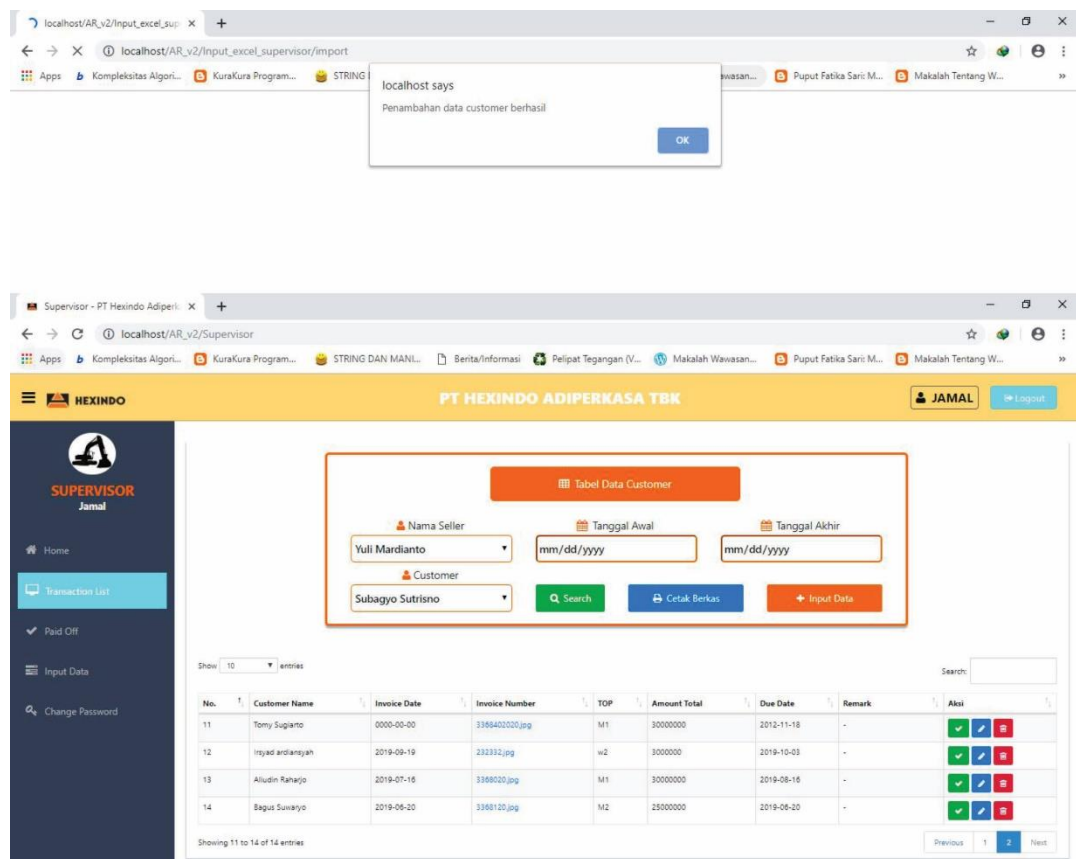
Setelah mengunggah *file excel*, *supervisor* dapat melihat tampilan *preview* dari data pada *file* tersebut. Hal ini berfungsi untuk membantu *supervisor* dalam memastikan kembali data-data yang akan dimasukkan ke dalam *database* merupakan data yang benar dan sesuai yang diharapkan. Tampilan *preview* dapat dilihat pada Gambar 4.11.



Gambar 4.11 Tampilan *preview file excel*

Setelah *supervisor* memastikan data yang dimasukkan adalah data yang benar, maka *supervisor* dapat langsung memasukkan data ke *database* dengan cara menekan tombol “import”. Apabila seluruh data berhasil masuk ke *database* maka akan muncul

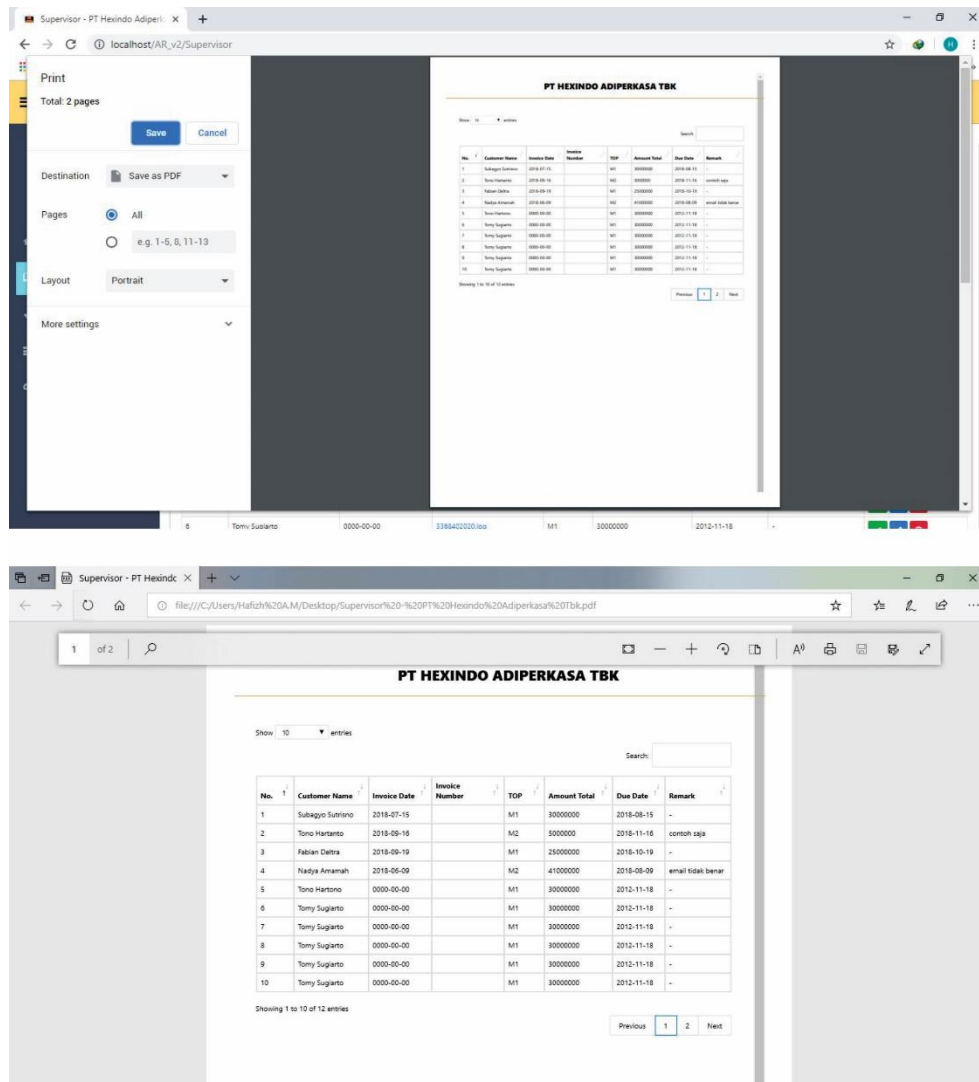
*pop up* yang bertuliskan “Penambahan data customer berhasil” dan akan langsung ditampilkan pada tabel *transaction list*. Tahapan ini dapat dilihat pada Gambar 4.12.



Gambar 4.12 Tampilan berhasil memasukkan data dari *file excel* (atas) dan *transaction list* setelah proses *input* data berhasil (bawah)

#### 4.2.7. Fitur Export to PDF

Pengujian terhadap fitur ini dilakukan dengan menekan tombol “Cetak Berkas” yang berada pada halaman *transaction list* dan *paid off* baik pada akun dengan *role supervisor* maupun *seller* dan akan menampilkan *pop up* untuk menyimpan berkas dalam bentuk format pdf. Setelah itu karyawan akan menekan tombol *save* dan *file* akan tersimpan di komputer atau laptop karyawan tersebut.



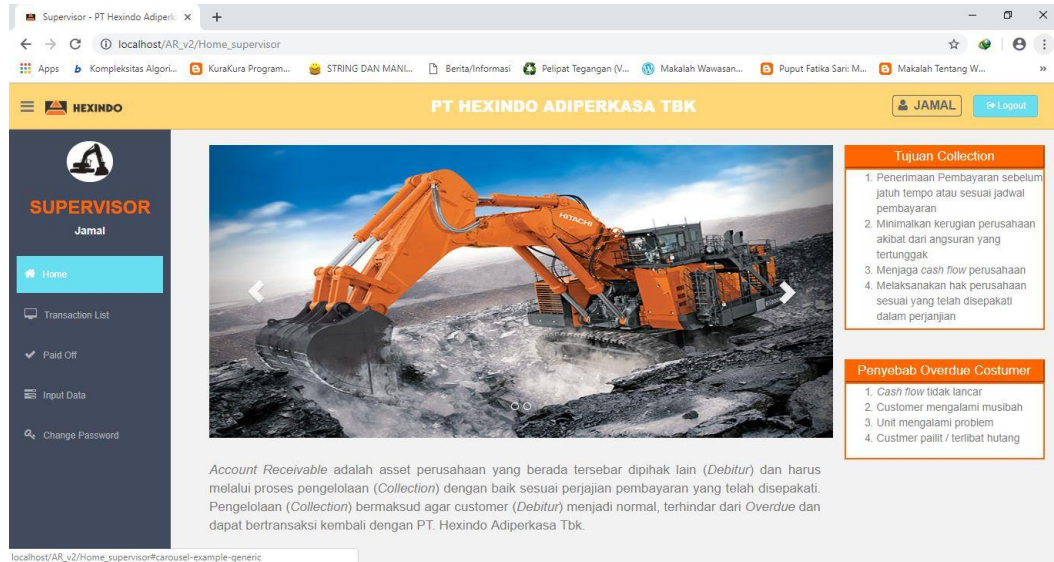
Gambar 4.13 Tampilan *export* PDF

Gambar 4.13 menunjukkan bahwa sistem mampu menghasilkan daftar pelanggan baik yang sudah maupun yang belum melunasi hutang dengan keluaran PDF. Daftar pelanggan tersebut juga dapat menyesuaikan dengan hasil pengurutan data maupun hasil *filter* data. Fitur ini membantu pihak perusahaan terutama karyawan untuk melaporkan daftar pelanggan yang sudah melunasi ataupun yang belum melunasi hutang pada rentan waktu tertentu dan menjadi bahan evaluasi kedepannya.

#### 4.2.8. Fitur *Dashboard*

Fitur berbentuk halaman yang dapat menampilkan foto dari berbagai jenis barang yang dijual oleh PT. Hexindi Adiperkasa, Tbk. Selain hal tersebut, pada halaman ini juga terdapat tujuan *collection* dan penyebab *overdue customer*, karena kedua hal inilah

yang terus ditekankan oleh pihak perusahaan kepada karyawan-karyawannya. Halaman ini yang menjadi halaman awal ketika akun dengan *role supervisor* atau *seller* berhasil melakukan *login* ke aplikasi. Tampilan halaman fitur ini dapat dilihat pada Gambar 4.14.



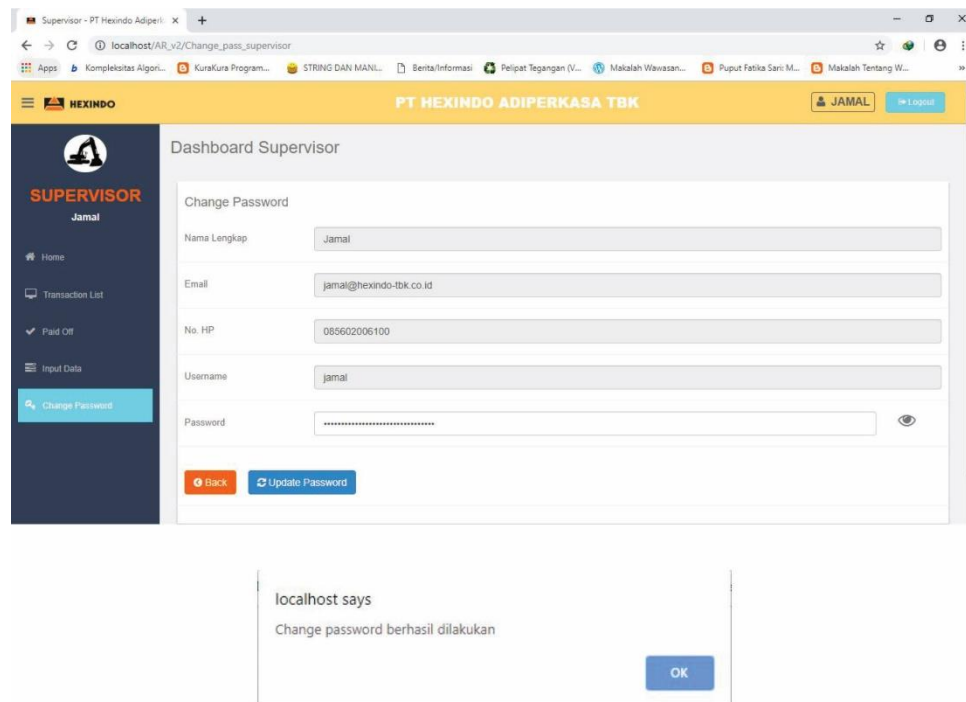
Gambar 4.14 Tampilan Daftar Barang

Berdasarkan Gambar 4.14, hasil pengujian terhadap fitur ini berhasil, karena halaman ini mampu menampilkan berbagai macam foto barang yang dijual oleh PT. Hexindo Adiperkasa Tbk. dalam bentuk *slide show* yang dapat digeser ke kanan maupun ke kiri. Selain hal tersebut, halaman ini juga berhasil menampilkan tujuan *collection* dan penyebab *overdue customer* yang dapat di *hide* dan dimunculkan kembali dengan efek javascript.

#### 4.2.9. Fitur Change Password

Pengujian terhadap fitur ini dilakukan dengan cara pengguna melakukan *login* pada aplikasi, lalu ke halaman “Change Password”. Setelah itu pengguna mengetikkan dari *password* baru yang diinginkan. Secara pengaturan awal, *password* yang diketikkan pengguna akan menjadi bulatan-bulatan hitam. Hal ini berfungsi guna mengamankan *password* dari penglihatan orang disekitarnya, namun apabila pengguna ingin memastikan apakah yang diketikkan merupakan yang diinginkan, pengguna dapat menekan *icon* mata yang berada di samping isian *password* untuk melihat hasil yang diketikkan.

Setelah itu pengguna hanya perlu menekan tombol “Update Password” untuk mereset *password*. Apabila perubahan *password* berhasil dilakukan, maka akan muncul *pop up* yang bertuliskan “Change password berhasil dilakukan”. *Session* dari pengguna tersebut dihapus dan akan kembali ke halaman *login*. Untuk tampilan dari pengujian ini dapat dilihat pada Gambar 4.15.

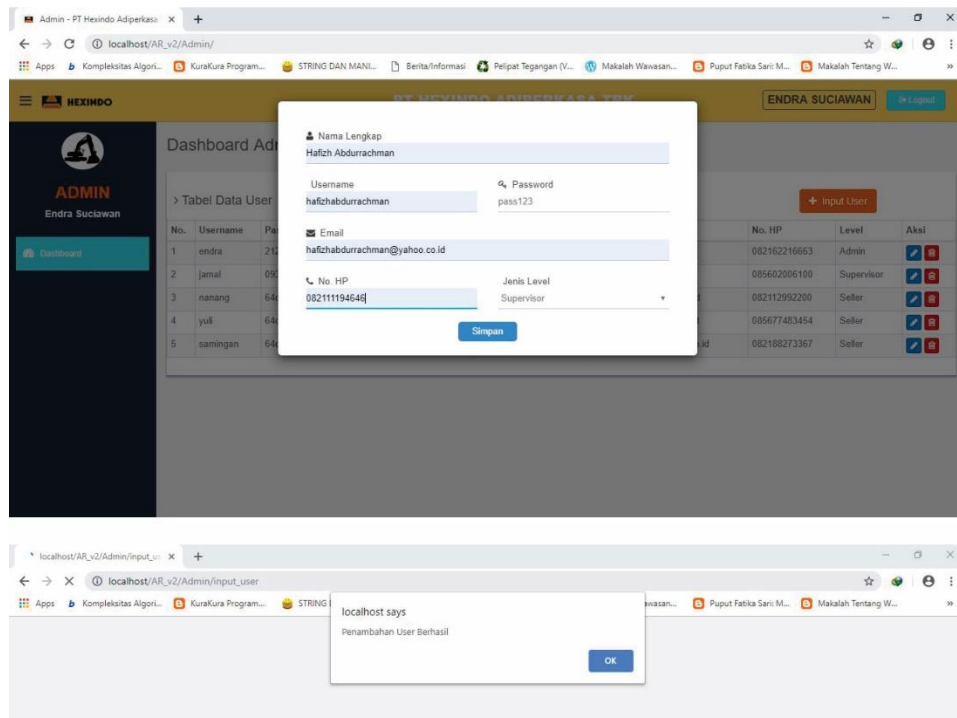


Gambar 4.15 Tampilan *change password* (atas) dan *pop up* berhasil mengubah *password*

#### 4.2.10. Fitur Menambah Akun Baru

Pengujian pada fitur ini dilakukan dengan cara admin menambahkan satu akun baru dengan *role* tertentu. Yang perlu diisikan oleh admin untuk membuat suatu akun baru adalah nama lengkap, *username*, *password*, *email*, nomor hp, dan *role*. Apabila berhasil menambah akun baru, akan muncul *pop up* yang bertuliskan “Penambahan user berhasil”. Tampilan penambahan akun dan *pop up* yang menandakan admin berhasil menambahkan akun baru dapat dilihat pada Gambar 4.16.

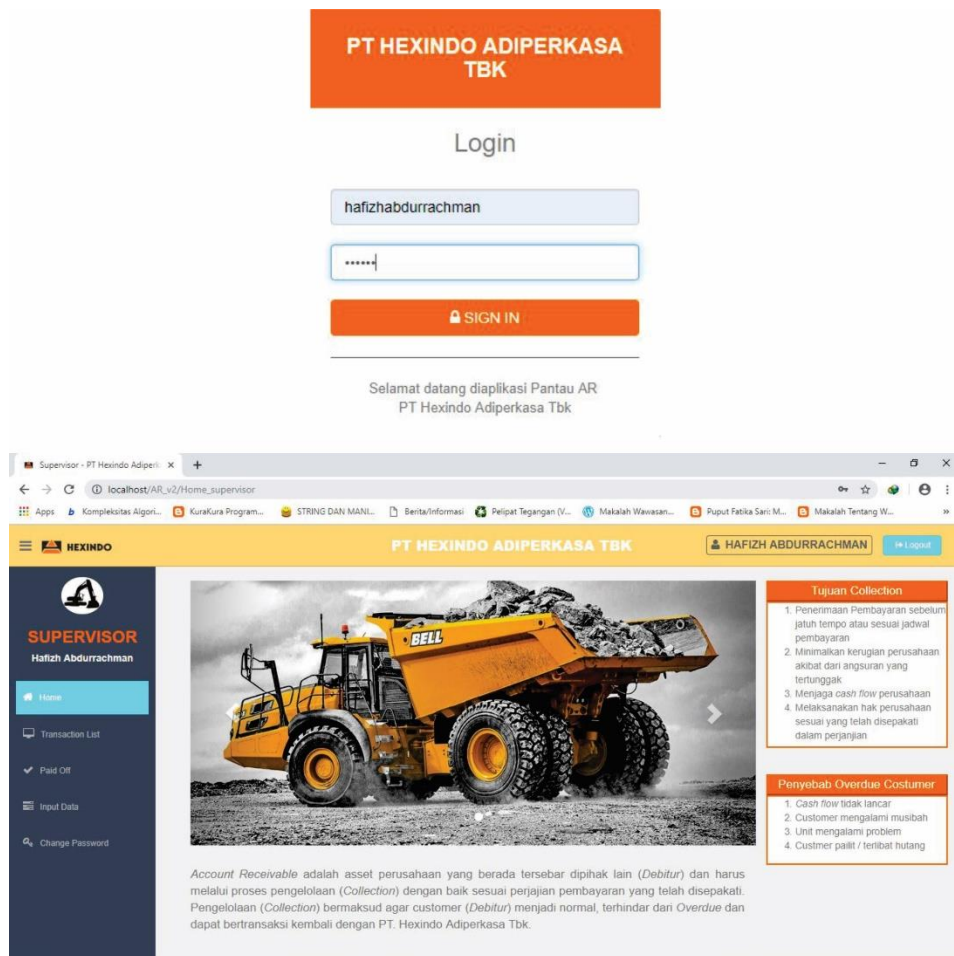




Gambar 4.16 Tampilan menambahkan akun baru (atas) dan *pop up* berhasil menambahkan akun baru

Setelah akun berhasil ditambahkan, akun tersebut akan secara otomatis aktif dan dapat melakukan *login* dengan *username* dan *password* yang telah dibuat seperti pada Gambar 4.17. Pengguna baru tidak perlu untuk melakukan aktivasi akun, karena pembuatan akun baru pengguna baik *seller* maupun *supervisor* sudah atas izin pihak perusahaan.





Gambar 4.17 Tampilan akun baru melakukan *login*

#### 4.2.11. Fitur Reminder Otomatis

Pengujian terhadap fitur ini dilakukan dengan mengarahkan alamat *email* ke alamat *email* pengembang dengan data yang dikirimkan merupakan data *dummy*. *Email* berhasil terkirim sesuai dengan yang diharapkan. Pembahasan terhadap fitur dan tampilan hasil dari pengujian fitur ini akan dibahas secara khusus pada Peningkat Waktu Otomatis.

#### 4.2.12. Evaluasi hasil pengujian black box

Evaluasi hasil akhir pengujian pada seluruh fitur aplikasi dengan metode *black box* dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil pengujian seluruh fitur dengan metode *black box*

No.	Nama Fitur Diuji	Kriteria Evaluasi Hasil	Hasil Keluaran	Kesimpulan
1.	Fitur <i>Login</i> dan <i>Multilevel User</i>	Sistem dapat menampilkan halaman <i>Dashboard</i> yang sesuai dari masing – masing <i>role</i> berdasarkan <i>username</i> dan <i>password</i> yang dimasukkan. Apabila <i>username</i> atau <i>password</i> yang dimasukkan tidak terdapat pada database, maka aplikasi akan memberikan pemberitahuan bahwa <i>username</i> atau <i>password</i> yang dimasukkan salah.	Sistem mengarahkan halaman yang benar pada setiap jenis <i>role</i> yang melakukam <i>login</i> . Apabila pengguna memasukkan <i>username</i> atau <i>password</i> yang salah, sistem menampilkan notif ” <i>username</i> atau <i>password</i> yang anda masukan salah”.	Berhasil
2.	Fitur <i>Transaction List (Supervisor)</i>	Sistem mampu menampilkan seluruh daftar pengguna yang belum melakukan pembayaran hutang. Sistem juga dapat mengurutkan data, melakukan filter terhadap data pengguna berdasarkan nama <i>seller</i> maupun nama pengguna, mengubah data pengguna dari <i>Transaction List</i> ke <i>Paid Off</i> , serta menghapus data tersebut.	Sistem menampilkan seluruh data pengguna yang belum melunasi pembayaran hutang.	Berhasil
3.	Fitur <i>Transaction List (Seller)</i>	Sistem mampu menampilkan seluruh daftar pengguna yang belum melakukan pembayaran hutang (yang menggunakan perantara <i>seller</i> tersebut saja). Sistem juga dapat mengurutkan data dan melakukan filter terhadap data pengguna berdasarkan nama pengguna (yang menggunakan perantara <i>seller</i> tersebut saja).	Sistem menampilkan seluruh data pengguna (yang menggunakan perantara <i>seller</i> tersebut saja) yang belum melunasi pembayaran hutang.	Berhasil

No.	Nama Fitur Diuji	Kriteria Evaluasi Hasil	Hasil Keluaran	Kesimpulan
4.	Fitur <i>Edit Data</i> ( <i>Supervisor</i> )	Sistem mampu menampilkan data pelanggan, mengubah data tersebut, serta menyimpannya kembali ke <i>database</i> .	<i>Supervisor</i> berhasil mengubah data dan menyimpannya kembali ke <i>database</i> .	Berhasil
5.	Fitur <i>Paid Off</i> ( <i>Supervisor</i> )	Sistem mampu menampilkan seluruh daftar pengguna yang sudah melakukan pembayaran hutang. Sistem juga dapat mengurutkan data, melakukan filter terhadap data pengguna berdasarkan nama <i>seller</i> maupun nama pengguna, serta menghapus data tersebut.	Sistem menampilkan seluruh data pengguna yang sudah melunasi pembayaran hutang.	Berhasil
6.	Fitur <i>Paid Off</i> ( <i>Seller</i> )	Sistem mampu menampilkan seluruh daftar pengguna yang sudah melakukan pembayaran hutang (yang menggunakan perantara <i>seller</i> tersebut saja). Sistem juga dapat mengurutkan data dan melakukan filter terhadap data pengguna berdasarkan nama pengguna.	Sistem menampilkan seluruh data pengguna (yang menggunakan perantara <i>seller</i> tersebut saja) yang sudah melunasi pembayaran hutang.	Berhasil
7.	Fitur <i>Input Data Manual</i> ( <i>Supervisor</i> )	Sistem mampu memasukkan satu data pelanggan dan menyimpannya ke dalam <i>database</i> .	<i>Supervisor</i> berhasil menambah satu data pelanggan.	Berhasil
8.	Fitur <i>Input Data Manual</i> ( <i>Seller</i> )	Sistem mampu memasukkan satu data pelanggan (yang menggunakan perantara <i>seller</i> tersebut saja). dan menyimpannya ke dalam <i>database</i> .	<i>Seller</i> berhasil menambah satu data pelanggan.	Berhasil
9.	Fitur <i>Input Data Excel</i> ( <i>Supervisor</i> )	Sistem mampu memasukkan beberapa data pelanggan sekaligus dengan format <i>file excel</i> yang sudah ditentukan dan menyimpannya ke dalam <i>database</i>	<i>Supervisor</i> berhasil menambah beberapa data pelanggan melalui <i>file excel</i> yang diunggah.	Berhasil

No.	Nama Fitur Diuji	Kriteria Evaluasi Hasil	Hasil Keluaran	Kesimpulan
10.	Fitur <i>Export to PDF</i> (Supervisor)	Sistem mampu mencetak daftar data pelanggan baik dalam tabel <i>Transaction List</i> maupun <i>Paid Off</i> pada <i>Dashboard Supervisor</i> dalam bentuk keluaran <i>PDF</i> .	Daftar pelanggan terunduh dengan frmat <i>PDF</i> .	Berhasil
11.	Fitur <i>Export to PDF</i> (Seller)	Sistem mampu mencetak daftar data pelanggan baik dalam tabel <i>Transaction List</i> maupun <i>Paid Off</i> pada <i>Dashboard Seller</i> dalam bentuk keluaran <i>PDF</i> .	Daftar pelanggan terunduh dengan frmat <i>PDF</i> .	Berhasil
12.	Fitur <i>Dashboard</i>	Sistem mampu menampilkan tujuan <i>collection</i> , dan penyebab <i>overdue customer</i> , serta foto berbagai jenis barang yang dijual oleh PT. Hexindo Adiperkasa, Tbk.	Menampilkan <i>slide show</i> berbagai jenis barang, tujuan <i>collection</i> , dan penyebab <i>overdue customer</i> .	Berhasil
13.	Fitur <i>Change Password</i>	Sistem mampu mengubah <i>password</i> ketika terdapat akun <i>seller</i> maupun <i>supervisor</i> sudah melakukan <i>login</i> dan ingin mengubah <i>password</i> akunnya sendiri.	<i>Password supervisor</i> maupun <i>seller</i> terubah.	Berhasil
14.	Fitur Menambah Akun Baru	Sistem mampu memberikan akses kepada pengguna dengan <i>role</i> admin untuk menambahkan akun <i>supervisor</i> atau <i>seller</i> baru.	Admin berhasil menambah akun baru dan berhasil melakukan <i>login</i> sesuai dengan <i>role</i> -nya.	Berhasil
15.	Fitur <i>Reminder</i> Otomatis	Sistem mampu mengirimkan <i>reminder</i> berupa <i>email</i> yang berisikan tagihan, <i>invoice</i> , tanggal tenggat waktu pembayaran, dan informasi pengguna secara otomatis apabila terdapat pelanggan yang sudah memasuki h-4 waktu tenggat pembayaran.	<i>Email</i> terkirim dengan data yang sesuai.	Berhasil

### 4.3 Pengingat Tenggat Waktu Otomatis

Pada bagian ini akan dipaparkan mengenai cara kerja pengingat tenggat waktu otomatis seperti pengaturan *cron job* pada hosting untuk melakukan penjadwalan pengiriman *email*, *back end script* fungsi *reminder*, dan bentuk dari *email* yang dikirimkan ke pengguna. Ketiga hal tersebut akan dipaparkan secara detail pada bagian ini.

#### 4.3.1. Pengaturan Cron Job pengingat tenggat waktu otomatis

Pada tahap ini *cron job* berfungsi sebagai pengatur jadwal pengiriman *email* pengingat pembayaran tenggat waktu otomatis. Pengaturan detail *cron job* dapat dilihat pada Gambar 4.18.



Gambar 4.18 Konfigurasi *cron job*

Terdapat 2 bagian penting pada pengaturan *cron job*, yaitu pengaturan waktu penjadwalan dan lokasi *file* serta fungsi yang akan dijalankan. Pada gambar diatas pengaturan waktu penjadwalan digambarkan dengan 00 10 \* \* \* di mana hal tersebut menunjukkan bahwa program akan dijalankan setiap hari pada jam 10 pagi. 00 10 menunjukkan program akan dijalankan pada menit ke 00 dan jam 10, serta ketiga bintang sisanya menunjukkan bahwa program dijalankan setiap hari.

Tulisan yang berwarna merah menunjukkan lokasi direktori *file* yang akan dieksekusi. Berdasarkan Gambar 4.18 lokasi *file* yang akan dieksekusi berada pada lokasi /usr/local/bin/php /home/pantauar/public\_html/index.php Cobaemail reminder. **Cobaemail** merupakan nama *file*, serta **reminder** merupakan fungsi yang akan dieksekusi sesuai dengan waktu pada *cron job*.

#### 4.3.2. Script back end fungsi pengingat tenggat waktu otomatis

a) Pencarian data pada *field reminder*

```
date_default_timezone_set('Asia/Jakarta');
$tanggal = date('d F Y');
$tgl=date('Y-m-d');
$sql= mysqli_query($connect,"SELECT * from penjualan");
while ($hasil = mysqli_fetch_array($sql)) {
    $reminder = $hasil['reminder'];
    if($reminder==$tgl){
        .....(1)
    }
    .....
}
```

Gambar 4.19 Script code reminder bagian 1

Pada tahap ini, program akan mencari seluruh data pada tabel penjualan yang memiliki tanggal waktu *reminder* sama dengan tanggal program dijalankan (program dijalankan setiap hari pada jam 10 sesuai dengan pengaturan *cron job*). Apabila terdapat data peminjaman yang memiliki tanggal *reminder* sama dengan tanggal program dijalankan, program akan melanjutkannya ke bagian (1) pada Gambar 4.19, namun apabila tidak ada, program akan melewati data tersebut dan beralih ke data selanjutnya. Pada *script* ini digunakan waktu *timezone* Kota Jakarta, sehingga waktu pembacaan program akan disesuaikan dengan waktu di Kota Jakarta. Hal ini dilakukan karena kantor pusat PT. Hexindo Adiperkasa Tbk. berada di Kota Jakarta, sehingga waktu pengiriman dan tanggal akan mengacu pada kota di mana kantor pusat berada.

b) Penginisialisasian data pelanggan dan pemanggilan fungsi *email*

```
$nama_peminjam = $hasil['nama_peminjam'];  
$email_peminjam = $hasil['email_peminjam'];  
$nama_seller = $hasil['nama_seller'];  
$email_seller = $hasil['email_seller'];  
$invoice_number = $hasil['invoice_number'];  
$deadline = $hasil['deadline'];  
$biaya = $hasil['biaya'];  
  
require "Exception.php";  
require "PHPMailer.php";  
require "SMTP.php";
```

Gambar 4.20 *Script code reminder* bagian 2

Pada tahap ini, penginisialisasian data peminjam dilakukan dengan memanfaatkan nilai data yang sudah diambil dari tabel penjualan. Penginisialisasian sangat penting dan diperlukan dalam memberikan nilai pada objek-objek tertentu, baik ketika mengatur konfigurasi *email* maupun pada *text body email*. Penginisialisasian ini juga berfungsi untuk memudahkan pengembang ketika ingin menelusuri nilai dari suatu variable.

Tahap selanjutnya adalah, pemanggilan *library* dari PHPMailer. *Library* dari *phpmailer* ini terdiri dari 3 *file* (Exception.php, PHPMailer.php, dan SMTP.php) yang disimpan pada *controller*. Ketiga *file* tersebut dibutuhkan dalam pengiriman *email* dan harus dipanggil dengan perintah *require*, apabila salah satu diantara ketiga *file* tersebut tidak dipanggil, maka fungsi pengiriman *email* tidak akan berjalan.

### c) Konfigurasi Email

```
$mail = new PHPMailer(true); // Passing `true` enables exceptions
try {
    //Server settings
    $mail->SMTPDebug = 2;           // Enable verbose debug output
    $mail->Mailer = 'smtp';         // Set mailer to use SMTP
    $mail->Host = 'pantau-ar.com';
    $mail->SMTPAuth = true;         // Enable SMTP authentication
    $mail->Username = 'support@pantau-ar.com'; // SMTP username
    $mail->Password = '*****';    // SMTP password
    $mail->SMTPSecure = 'ssl';
    $mail->Port = 465;              // TCP port to connect to

    //Recipients
    $mail->setFrom('support@pantau-ar.com', 'PT Hexindo Adiperkasa Tbk');
    $mail->addAddress($email_peminjam,$nama_peminjam); // Add a recipient
    $mail->addAddress($email_seller,$nama_seller);    // Add a recipient
    $mail->addReplyTo('support@pantau-ar.com');

    //Content
    $mail->isHTML(true);            // Set email format to HTML
    $mail->Subject = "Konfirmasi Pembayaran";
    $mail->Body = ".....(d)";
    $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';
    $mail->send();
} catch (Exception $e) {
    echo 'Message could not be sent. Mailer Error: ', $mail->ErrorInfo;
}
```

Gambar 4.21 *Script code reminder* bagian 3



Tahap ini merupakan tahap mengatur konfigurasi *email*. Terdapat banyak hal yang harus diatur seperti *Host*, *username*, *password*, *port*, *email* pengirim, *email* penerima, *subject email*, isi *email*, dan lain-lainnya. Pembahasan akan dibagi menjadi 3 bagian, yaitu :

- *Server Setting*

Pada bagian ini alamat host disesuaikan dengan nama domain, yaitu “pantau-ar.com”. *username* dan *password* menyesuaikan dengan *username* dan *password* ketika membuat *email* pada hosting. Untuk SMTPsecure dan port yang digunakan, disesuaikan dengan saran dari pihak hosting, di mana yang disarankan adalah *ssl* dengan port 465, namun tidak semuanya seperti ini, terkadang ada yang disarankan *tsl* dengan port 587. Lalu, konfigurasi lainnya seperti SMTPDebug, Mailer, dan SMTPAuth disamakan seperti konfigurasi dasar bawaan dari *phpmailer*.

- *Recipient*

Pada bagian ini terdapat 3 bagian penting yaitu *setfrom* yang digunakan untuk mengatur *email* yang akan digunakan dalam pengiriman pengingat tenggat waktu otomatis (d disesuaikan dengan *username* dan *password* ketika mengatur konfigurasi server), *addAddress* merupakan *email* target (bisa dikirimkan ke lebih dari satu alamat setiap pengiriman *email*) yang akan dikirimkan *email* (mengambil dari penginisialisasian variable yang sudah dijelaskan sebelumnya), dan yang terakhir adalah *addReplyTo* yang digunakan sebagai *email* tujuan apabila target ingin membalas *email* pengingat tenggat waktu pembayaran otomatis ini.

- *Content*

Bagian ini merupakan bagian isi dari *email*. Yang perlu diatur pada bagian ini cukup *Subject* dan *Body* saja, disesuaikan dengan permintaan perusahaan. Untuk bagian *Body* diisi kode HTML dan CSS agar bentuk *email* lebih menarik. Selain itu, *body email* ini juga dapat disisipkan variable PHP yang telah diinisialisasikan diawal untuk memberikan data pengguna seperti nama pengguna, *invoice number*, waktu tenggat pembayaran, dan total biaya.

d) Isi *body email*

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
</head>
<body style='margin-left: 20px;border:2px solid;'+
  <section style='margin-left: 5%;'+
    <table style='width: 100%;'+
      <tr>
        <td rowspan='3'><img src='http://paabi.id/wp-content/uploads/2015/09/Logo-
Hexindo-horisontal.png' style='width: 200px;height: 50px;'+></td>
        <td><br></td>
      </tr>
      <tr>
        <td><br></td>
      </tr>
      <tr>
        <td style='padding-left: 15%;'+>Banjarmasin, <b>$tanggal</b></td>
      </tr>
    </table>
    <div style='margin-top: 3%;'+
      Kepada :<br>
      <b>$nama_peminjam</b><br>
      <b>Di Tempat</b>
    </div>
    <div style='margin-top: 3%;'+
      Perihal : Konfirmasi Piutang
    </div>
    <div style='margin-top: 1%;text-align: justify;margin-right: 7%;'+
      &nbsp;&nbsp;&nbsp;Terlampir kami sampaikan bahwa saldo hutang Bapak/Ibu kepada kami atas
pembelian spare part dan jasa service dengan No Invoice <b>$invoice_number</b> yang akan jatuh tempo
pada tanggal <b>$deadline</b> adalah sebesar <b>$biaya</b>. Mohon dapat segera melunasi piutang
tersebut sebelum waktu jatuh tempo yang telah ditentukan.
      <br><br>
      Pembayaran dapat di transfer ke Rekening kami
    </div>
    <div style='margin-top: 2%;margin-left: 5%;'+
      PT. Hexindo Adipekasa Tbk
      <br><br>
      Bank Mandiri cabang Banjarbaru
      <br><br>
      A/C 031.00.0994595.0 (IDR)
    </div>
    <div style='margin-top: 2%;text-align: justify;margin-right: 7%;'+
      &nbsp;&nbsp;&nbsp;Untuk informasi lainnya, silahkan hubungi customer support kami dengan
nomor telepon : 0511-4706056 dengan Sdr. Jamal (Service Section Head) atau Sdr Nanang Ari Sanjaya (PSR).
    </div>
    <div style='margin-top: 2%;text-align: justify;margin-right: 7%;'+
      &nbsp;&nbsp;&nbsp;Namun apabila Bapak/Ibu telah melakukan pembayaran tagihan tersebut,
kami mengucapkan terimakasih dan mohon abaikan surat ini.
    </div>
    <table style='margin-top: 3%;margin-left: 60%;text-align: center;'+
      <tr>
        <td>PT. Hexindo Adiperkasa Tbk<br><br><br><br><br></td>
      </tr>
      <tr>
        <td>Catur Busono Riyadi</td>
      </tr>
    </table>
  </section>
</body>
</html>
```

Gambar 4.22 *Script code body email reminder*

Gambar 4.22 menunjukkan *script code email* yang aka dikirimkan ke pelanggan sebagai pengingat tenggat waktu pembayaran. *Script code* tersebut hanya terdiri dari HTML, CSS, dan PHP. HTML berfungsi dalam mengatur kerangka *email* yang dikirimkan sehingga tampilannya akan menyerupai surat formal, bukan sebatas *text* biasa. CSS berfungsi dalam memperindah kerangka *email* yang sudah dibangun menggunakan HTML seperti tulisan menjadi rata tengah, pengaturan batas atas, bawah,

kanan, dan kiri, dan border *email*. PHP berfungsi dalam memberikan nilai berdasarkan data yang telah diinisialisasikan sebelumnya pada *email*. Beberapa contoh beberapa variabel yang digunakan pada *email* diatas adalah \$nama\_peminjam, \$invoice\_number, dan \$deadline. Hal ini lah yang membuat *email* yang dikirimkan sesuai dengan data pelanggan yang berada pada basis data.

#### 4.3.3. Tampilan email pengingat tenggat waktu otomatis

Pada bagian ini akan dibahas mengenai tampilan *email* pengingat tenggat waktu pembayaran yang dikirimkan kepada pelanggan. Tampilan dari *email* dapat dilihat pada Gambar 4.23.



Gambar 4.23 Tampilan *email reminder* yang diterima oleh pengguna

Tampilan *email* diatas merupakan hasil dari *script code* pada Gambar 4.22. *Email* yang dikirimkan ke pelanggan sudah disesuaikan dengan data pelanggan tersebut, seperti nama pelanggan, tanggal pengiriman, nomor *invoice*, tanggal tenggat waktu pembayaran, dan total biaya hutang. Selain hal tersebut, dapat dilihat bahwa email pengirimnya adalah support@pantau-ar.com, di mana *email* tersebut sudah dikonfigurasi pada pengaturan *email* yang sudah dipaparkan di atas. Apabila

pelanggan ingin membalas *email* tersebut, maka alamat tujuan sudah otomatis terisi dengan alamat yang sama ([support@pantau-ar.com](mailto:support@pantau-ar.com)) karena sudah diatur pada bagian konfigurasi *email*. *Email* ini berfungsi sebatas sebagai pengingat kepada pelanggan agar segera membayarkan hutangnya melalui rekening yang tertera pada isi *email*.

#### **4.4 Perbandingan sistem baru terhadap sistem lama**

Pada bagian ini akan dipaparkan mengenai kelebihan, kekurangan, serta perbedaan dari sistem baru yang dikembangkan pada tugas akhir ini dengan sistem lama.

##### **4.4.1. Kelebihan Sistem**

Aplikasi pengingat tenggat waktu pembayaran ini memiliki beberapa kelebihan dari sistem sebelumnya di antaranya adalah :

1. Sistem dapat diakses di mana saja dan kapan saja karena terhubung dengan internet.
2. Seluruh data pelanggan terpusat dalam satu tempat penyimpanan *database*.
3. Pengingat tenggat waktu pembayaran untuk setiap pelanggan sudah dilakukan secara otomatis.
4. Penggunaan data sudah terbagi sesuai dengan hak akses masing – masing karyawan.
5. Penambahan data dapat dilakukan secara masal dengan mengunggah *file excel*.
6. Pelaporan daftar pelanggan dapat di *export* ke dalam bentuk PDF.
7. Terdapat berbagai jenis foto barang yang dijual.

##### **4.4.2. Kekurangan Sistem**

Adapun aplikasi ini masih memiliki kekurangan yang dapat dikembangkan selanjutnya seperti :

1. Sistem tidak dapat diakses apabila *device* tidak terhubung dengan internet.
2. Diperlukan biaya tambahan untuk sewa *hosting* dan *domain* pertahunnya.
3. Tampilan belum sepenuhnya *responsive*, sehingga beberapa halaman akan terlihat tidak rapih apabila dibuka melalui *mobile*.

#### 4.4.3. Perbedaan cara kerja sistem baru dan sistem lama

Pada bagian ini akan dipaparkan seluruh perbedaan cara kerja dari sistem lama (Microsoft Excel) ke sistem baru (aplikasi berbasis *website*). Perbedaan ini dapat dilihat pada Tabel 4.4 Perbedaan sistem baru dan sistem lama

Tabel 4.4 Perbedaan sistem baru dan sistem lama

No.	Aktivitas	Sstem Lama (Microsoft Ecel)	Sistem Baru (Aplikasi berbasis <i>website</i> )
1.	Melihat / mengakses data	Mendatangi komputer HO ( <i>Head Office</i> ) yang menjadi tempat penyimpanan pusat <i>file excel</i> ( <i>offline</i> ).	Membuka <i>website</i> dan <i>login</i> menggunakan akun masing-masing karyawan di mana saja dan kapan saja selama terhubung ke internet ( <i>online</i> ).
2.	Menambah data pelanggan baru	Memberikan data ke <i>admin</i> pusat dengan <i>file excel</i>	Menambahkan dengan fitur tambah data atau dengan mengunggah <i>file excel</i> (khusus <i>supervisor</i> )
3	Mengubah data pelanggan	Mengubah data pelanggan pada <i>file excel</i> atas izin <i>admin</i>	Menekan tombol <i>icon</i> edit dan mengganti data pelanggan melalui <i>form</i> yang disediakan (khsus <i>supervisor</i> )
4.	Menghapus data pelanggan	Menghapus data pelanggan pada <i>file excel</i> atas izin <i>admin</i>	Menekan tombol <i>icon delete</i> pada baris data pelanggan yang ingin dihapus (khsus <i>supervisor</i> )
5.	Pelaporan daftar pelanggan dalam kurun waktu tertentu	Meng-copy data pelanggan berdasarka kurun waktu tertentu pada <i>file excel</i> ke <i>file excel</i> baru, lalu dicetak	Menggunakan fitur <i>filter</i> rentan waktu, lalu menekan tombol cetak berkas
6	Mengingatkan tenggat waktu pembayaran hutang kepada pelanggan	Karyawan membuat surat pengingat tenggat waktu pembayaran dengan Microsoft Word, lalu mengirimkan <i>file</i> tersebut ke alamat pelanggan satu persatu.	Otomatis terkirim h-4 tenggat waktu pembayaran kepada masing-masing alamat <i>email</i> pelanggan
7.	Penyimpanan data pelanggan	Data disimpan pada <i>file excel</i> pusat pada bagian HO	Data disimpan pada <i>hosting</i> secara terpusat

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut :

1. Penelitian ini berhasil mengembangkan aplikasi sistem informasi berbasis *website* yang dapat membantu karyawan PT. Hexindo Adiperkasa Tbk. dalam
  - a. Menampilkan data pelanggan baik yang sudah maupun yang belum membayar hutang secara *online* sehingga dapat diakses di mana saja dan kapan saja selama terhubung dengan internet.
  - b. Mengintegrasikan dan memusatkan data penjualan pada satu penyimpanan basis data sehingga mengurangi adanya duplikasi data.
  - c. Mencari, menyaring, serta mengurutkan data penjualan.
  - d. Memanipulasi dan memodifikasi data penjualan.
  - e. Membatasi hak akses data pelanggan sesuai dengan jabatan karyawan.
  - f. Mencetak daftar pelanggan dalam keluaran PDF.
  - g. Mengingatkan karyawan terhadap pengenalan untuk setiap produk, tujuan *collection*, dan penyebab *overdue customer*.
2. Aplikasi ini mampu mengirimkan pengingat tenggat waktu pembayaran hutang secara otomatis kepada seluruh pelanggannya sehingga mengurangi biaya, waktu, serta tenaga yang dikeluarkan oleh karyawan dalam melakukan pengiriman surat tagihan pembayaran secara manual.
3. Pengujian terhadap seluruh fitur aplikasi menggunakan metode pengujian *black box* berjalan sesuai yang diharapkan dengan presentasi keberhasilan 100%
4. Dengan aplikasi sistem informasi ini, pengguna dapat dengan mudah melakukan proses pengolahan data pembayaran hutang pelanggan, sehingga proses perekapan data menjadi lebih cepat.

## 5.2 Saran

Aplikasi ini masih memiliki beberapa kekurangan. Beberapa saran yang dapat dipertimbangkan sebagai solusi dari kekurangan aplikasi ini adalah sebagai berikut :

1. Mengembangkan tampilan antarmuka aplikasi agar lebih *responsive* ketika diakses melalui berbagai jenis *device*.
2. Mengembangkan analisis performa kinerja karyawan berdasarkan jumlah pelanggan yang didapat berdasarkan kurun waktu tertentu.
3. Memperhatikan aspek performa kecepatan mengakses data sehingga proses pengolahan data dapat dilakukan lebih maksimal.

## DAFTAR PUSTAKA

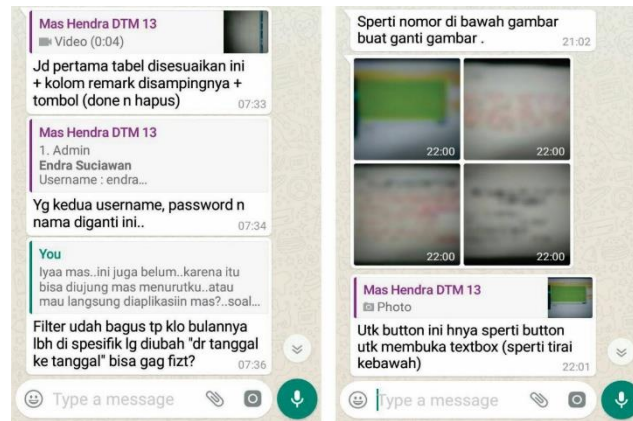
- [1] Hexindo, "About Hexindo," 2017. [Online]. Available: <https://www.hexindo-tbk.co.id/company/about-hexindo/>. [Accessed 27 August 2018].
- [2] H. Pratyaksa, "Perancangan Sistem Penjadwalan Otomatis Memanfaatkan Google Calender Berbasis Perangkat Plugwise," UGM, Yogyakarta, 2018.
- [3] A. Hernowo, "Perancangan Dan Implementasi Sistem Penjadwalan Servis Berkala Kendaraan Bermotor Berbasis Website menggunakan Notifikasi SMS Gateway," Universitas Kristen Satya Wacana, Salatiga, 2016.
- [4] G. Ariyanto, "Sistem Informasi Pengingat Jadwal General Check Up Di Laboratorium Prima Medika Kudus Berbasis Sms Gateway & Email," Universitas Muhammadiyah Surakarta, Solo, 2015.
- [5] H. M. A. N. A. Choirul, "Sistem Reminder Untuk Pemberitahuan Masa Jatuh Tempo Pembayaran Secara Otomatis Di Ksp Bhakti Karya Magelang," Universitas Muhammadiyah Magelang, Magelang, 2018.
- [6] Developer-Mozilla, "HTML Basic," 2018. [Online]. Available: [https://developer.mozilla.org/enUS/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/enUS/docs/Learn/Getting_started_with_the_web/HTML_basics). [Accessed 24 february 2019].
- [7] TechTerms, "CSS," [Online]. Available: <https://techterms.com/definition/css>. [Accessed 24 February 2019].
- [8] Techopedia, "JavaScript (JS)," 2017. [Online]. Available: <https://www.techopedia.com/definition/3929/javascript-js>. [Accessed 24 February 2019].
- [9] Duniaikom, "Index Tutorial Belajar MySQL Duniaikom," 2017. [Online]. Available: <https://www.duniaikom.com/tutorial-belajar-mysql-dan-index-artikel-mysql/>. [Accessed 7 August 2018].
- [10] PHP, "What is PHP?," [Online]. Available: <https://www.php.net/manual/en/intro-what-is.php>. [Accessed 24 February 2019].
- [11] A. Hadi, "Pengertian dan Cara Menggunakan CodeIgniter," 2016. [Online]. Available: <https://www.malasngoding.com/pengertian-dan-cara-menggunakancodeigniter/>. [Accessed 16 August 2018].



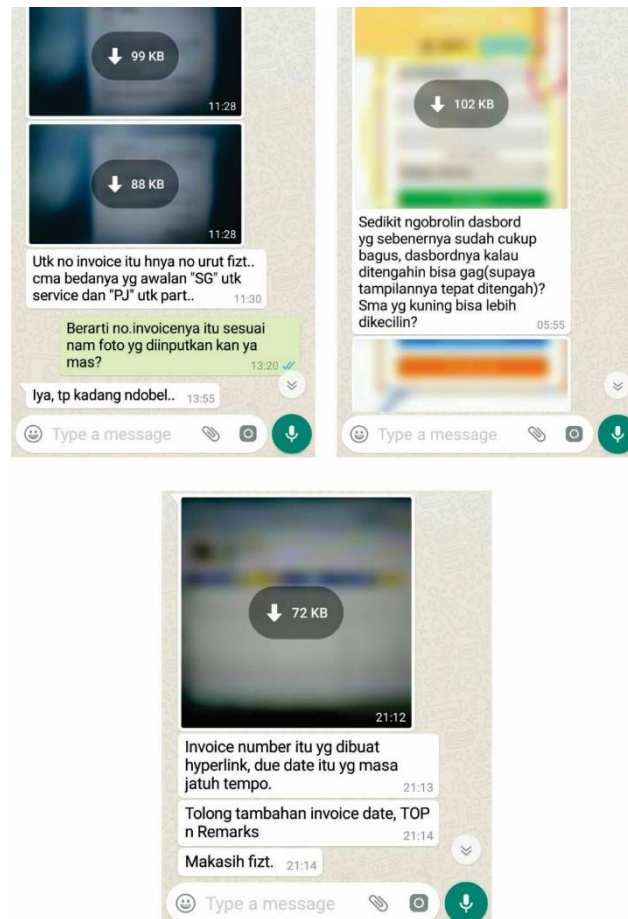
- [12] Techopedia, "Scheduler," 2017. [Online]. Available: <https://www.techopedia.com/definition/25078/scheduler>. Diakses 9 Februari 2019. . [Accessed 9 February 2019].
- [13] J. E. K. Kenneth E. Kendall, "Rapid Application Development," in *SYSTEM ANALYSIS AND DESIGN 8 Edition*, Camden, New Jersey, S4Carlisle Publishing Service, Inc., 2011, pp. 163-164.
- [14] Lucidchart Content Team, "4 Phases of Rapid Application Development Methodology," 23 May 2018. [Online]. Available: <https://www.lucidchart.com/blog/rapid-application-development-methodology>. [Accessed 21 April 2019].
- [15] R. E. Don Gosselin, "PHP," in *PHP Programming with MySQL Approach 2th Edition*, Boston , Course Technology, 2011, p. XIV.
- [16] Website, "Web Hosting," [Online]. Available: <https://www.website.com/beginnerguides/webhosting/6/1/what-is-webhosting?.ws>. [Accessed 21 March 2019].
- [17] M. A. v. d. Linden, "Black-Box Versus White-Box Security Testing," in *Testing Code Security*, Boca raton, Florida, Auerbach Publications Taylor & Francis Group, 2017, p. 61.
- [18] Website, "Black Box Testing," [Online]. Available: <http://softwaretestingfundamentals.com/blackbox-testing/>. [Accessed 21 April 2019].

## LAMPIRAN

**Lampiran 1. Masukan dari pihak PT. Hexindo Adiperkasa Tbk. dalam proses iterasi pengembangan aplikasi.**



**Lampiran 2. Masukan dari pihak PT. Hexindo Adiperkasa Tbk. dalam proses iterasi pengembangan aplikasi.**



### Lampiran 3 Source code *Login Multilevel User*

```
function do_login() {
    include 'Koneksi.php';
    $username = $this->input->post('username');
    $password = md5($this->input->post('password'));
    $cek = $this->M_login->cek_user($username, $password);
    if (count($cek) > 0 ) {
        $sql = mysqli_query($connect,"SELECT * from user where username
= '$username' and password = '$password' ");
        while ($kolom = mysqli_fetch_array($sql)) {
            $cek_type = $this->db->get_where('user', array('username'=>
$username,'password'=>$password));
            foreach ($cek_type->result() as $cek) {
                if ($cek->level=='Admin') {
                    $this->session->set_userdata(array(
                        'isLogin' => TRUE, //set data telah login atau belum
                        'username' => $username, //set session username
                        'password' => $password,
                        'nama' => $cek->nama,
                        'email' => $cek->email,
                        'hp' => $cek->hp,
                        'level' => $cek->level,
                    ));
                    redirect('Admin');
                }
                ..... (x)
            }
        }
    }
    else {
        $this->session->set_flashdata('gagallogin','Maaf,      Username
atau Password yang anda masukkan salah');
        $this->load->view('login');
    }
}
```