**Pengembangan Sistem Informasi Pemasaran pada Penerbit UGM Press**

**C-450**



**Disusun oleh:**

**Andrew Mulya 17/410461/TK/45818**

**Gian Willy Erlanda 17/410471/TK/45828**

**Michael Roberto Kristijanto 17/413904/TK/46344**

**DOKUMENTASI SKRIPSI *CAPSTONE PROJECT***

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA**

**2021**

# HALAMAN PENGESAHAN

|  |
| --- |
| 1. USULAN JUDUL SKRIPSI **: Pengembangan Sistem Informasi Pemasaran pada**  **Penerbit UGM Press**  **: The Development of Marketing and Royalty Reporting**  **Information System at UGM Press Publisher** |
| 2. JENIS DOKUMEN  **:** IMPLEMENTASI SOLUSI |
| 3. KODE DOKUMEN  **:** C450 |
| 4. NOMOR DOKUMEN **:** C450\_IWMPIS\_20\_2F539 |
| 5. NOMOR REVISI  **:** 00 |
| 6. TANGGAL PENERBITAN  **:** |
| 7. KETUA TIM Tanda Tangan :  a. Nama lengkap : Michael Roberto Kristijanto  b. NIM : 17/413904/TK/46344  c. Prodi/Konsentrasi : Teknologi Informasi/Rekayasa Perangkat Lunak  d. Email : michaelroberto99@mail.ugm.ac.id |
| 8. ANGGOTA 1 Tanda Tangan :  a. Nama lengkap : Gian Willy Erlanda  b. NIM : 17/410471/TK/45828  c. Prodi/Konsentrasi : Teknologi Informasi/Rekayasa Sistem Informasi  d. Email : gian.willy.erlanda@mail.ugm.ac.id |
| 9. ANGGOTA 2 Tanda Tangan :  a. Nama lengkap : Andrew Mulya  b. NIM : 17/410461/TK/45818  c. Prodi/Konsentrasi : Teknologi Informasi/Rekayasa Sistem Informasi  d. Email : andrewmulya@mail.ugm.ac.id |
| 10. DOSEN PEMBIMBING I Tanda Tangan :  a. Nama lengkap : I Wayan Mustika, S.T., M.Eng., Ph.D.  b. NIP : 198109212014041001 |
| 11. DOSEN PEMBIMBING II Tanda Tangan :  a. Nama lengkap : P. Insap Santosa, Ir., M.Sc., Ph.D., IPU.  b. NIP : 196101081985031002 |
| 12. TEMPAT PELAKSANAAN : UGM Press |
| 13. Jumlah halaman : |

# DAFTAR ISI

[**HALAMAN PENGESAHAN**](#_wx8nzfpb7af0) **2**

[**DAFTAR ISI**](#_zn7mxvitlkg) **3**

[**CATATAN REVISI DOKUMEN**](#_5m4nwx78v5lp) **6**

[INTISARI](#_30j0zll) **7**

[**RINGKASAN EKSEKUTIF**](#_3ix8f02cdvds) **9**

[PENDAHULUAN](#_1fob9te) **10**

[**PROSES DESAIN DAN IMPLEMENTASI**](#_owli0221podt) **11**

[Proses Desain Back-End](#_qxcvc79mt48p) 13

[Proses Desain Front-End](#_aj47oreqaoqy) 14

[Proses Desain User Experience](#_jt9f8cz9e9s7) 15

[Halaman Faktur](#_8802pa6lfhhl) 15

[Halaman Royalti](#_4bv5pbj7n0pe) 18

[Implementasi Fitur Customer dan Diskon](#_4tw3i87bvex2) 20

[Implementasi Fitur Faktur](#_n0qyaohu1by0) 24

[Implementasi Fitur Proforma](#_jcwtylpi3nh2) 24

[Implementasi Fitur Pendapatan](#_5wo8a4hgogxg) 24

[Implementasi Fitur Royalti](#_113o7vrbsyat) 24

[**PERUBAHAN PADA PROSES DESAIN DAN IMPLEMENTASI**](#_o0haj547zfmc) **25**

[**PENGUJIAN DAN ANALISIS**](#_pucklpa7n28n) **26**

[**KESIMPULAN**](#_gu6b6gn6defp) **58**

[**REFERENSI**](#_3ste3hyi5ltw) **59**

[**LAMPIRAN-LAMPIRAN**](#_53t3mnyxh291) **60**

[**REFERENSI (Bibliografi)**](#_55usv6d0fgg0) **61**

[**LAMPIRAN-LAMPIRAN**](#_ipo59iyjaysn) **63**

[**LAMPIRAN L-1**](#_wyhjgr2epeo2) **64**

# 

# CATATAN REVISI DOKUMEN

|  |  |  |  |
| --- | --- | --- | --- |
| **Versi** | **Tanggal** | Oleh | **Perbaikan** |
| a |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# INTISARI

UGM Press memiliki sistem informasi percetakan yang bernama SIGAP, yang digunakan oleh karyawan untuk keperluan operasional mereka. Namun dalam pelaksanaan proses bisnis UGM Press, proses pemasaran masih dilakukan secara konvensional. Proses pemasaran ini meliputi pembuatan faktur, perekapan penjualan, dan penghitungan royalti. Untuk mempermudah proses bisnis dalam pemasaran UGM Press agar menciptakan sistem yang terotomasi, dilakukan pengembangan SIGAP dengan menambahkan menu baru bernama Pemasaran yang meliputi fitur Proforma, Faktur, Showroom, Pendapatan, dan Royalti. Untuk menguji sistem pemasaran dalam SIGAP, dilakukan testing terhadap karyawan bagian pemasaran UGM Press sebagai calon pengguna sistem. Pengujian dilakukan dengan menggunakan metode whitebox testing untuk bagian back-end, blackbox testing untuk bagian front-end, dan System Usability Scale (SUS) untuk bagian user experience sebagai tingkat acuan keberhasilan fungsionalitas sistem.

# 

# RINGKASAN EKSEKUTIF

Universitas Gadjah Mada memiliki Unit Penunjang Universitas yang bergerak di bidang penerbitan dan percetakan yang bernama UGM Press. UGM Press memiliki sistem informasi yang bernama SIGAP (Sistem Informasi Gadjah Mada Press) untuk membantu melakukan proses bisnis di UGM Press, akan tetapi proses bisnis untuk bagian pemasaran belum terintegrasi oleh sistem.

Proses bisnis pemasaran meliputi pembuatan faktur, perekapan penjualan, dan penghitungan royalti. Untuk mengembangkan sistem informasi SIGAP, kami menambahkan beberapa fitur baru untuk memudahkan dalam menjalankan kegiatan pemasaran di UGM Press. Fitur baru yang dibuat meliputi Proforma, Faktur, *Showroom*, Pendapatan, dan Royalti, yang tergabung dalam satu menu yaitu Pemasaran. Proforma merupakan dokumen sementara sebagai alat untuk menunjukkan estimasi harga pembelian, dan proforma dapat berubah menjadi faktur oleh sistem apabila calon pembeli menyetujui dan membeli yang disepakati sebelumnya. Fitur faktur dan *showroom* berfungsi mengelola dan membuat dokumen yang sah sebagai bukti adanya transaksi jual beli di UGM Press. Fitur royalti memungkinkan adanya perhitungan pembagian keuntungan secara otomatis kepada pemilik buku yang bukunya terjual di UGM Press. Kemudian ditambahkan fitur menambah dan mengedit *customer* guna keperluan pemasaran dalam proses pembuatan faktur.

Sistem informasi SIGAP yang terbaru telah kami uji terhadap calon pengguna yaitu karyawan pemasaran UGM Press dan dilakukan sejumlah *testing* untuk simulasi penggunaan sistem. Hasil dari uji coba tersebut bernilai memuaskan dan calon pengguna sistem mudah untuk mempelajari sistem dan tidak menemukan adanya kesulitan.

# PENDAHULUAN

UGM Press adalah salah satu Unit Penunjang Universitas yang berada di bawah Badan Penerbit dan Publikasi Universitas Gadjah Mada (BPP UGM). Melalui SK Rektor UGM No. UGM/40/P/C pada tanggal 30 Juni 1971. Percetakan yang didirikan oleh Fakultas Ilmu Sosial dan Politik UGM pada tahun 1960 ini diresmikan dengan nama Gadjah Mada *University Press* atau UGM Press. UGM Press bergerak di bidang penerbitan dan percetakan di wilayah UGM. Perkembangan UGM Press sangat pesat, terbukti dengan menjadi salah satu penerbit perguruan tinggi di Indonesia yang diakui oleh *Southeast Asia University* dilihat dari jumlah buku yang diterbitkan. Terhitung dari tahun 1971 sampai 2015, sebanyak 2.000 judul buku telah diterbitkan UGM Press [1].

UGM Press mempunyai sistem informasi yang bernama Sistem Informasi Gadjah Mada Press (SIGAP). Sistem informasi ini digunakan untuk membantu proses bisnis UGM Press. SIGAP sendiri telah digunakan untuk mengotomatisasi proses bisnis pada tahap pra-produksi (pembuatan naskah, *review*, editorial, desain *cover*, dan *layout*), produksi (menerima order cetak, melakukan pencetakan, konfirmasi percetakan selesai), dan pasca-produksi (pemasaran) sehingga proses bisnis dapat dilakukan lebih cepat dan efisien.

Namun, proses pemasaran di UGM Press masih dilakukan secara konvensional. Proses pemasaran meliputi pembuatan faktur, perekapan data penjualan, dan penghitungan royalti. Proses ini membutuhkan waktu yang cukup lama sehingga proses bisnis kurang efektif jika dilakukan secara manual. Untuk mengatasi masalah ini, tim capstone kami akan mengembangkan SIGAP dengan melakukan penambahan fitur pemasaran dan penghitungan royalti secara otomatis di SIGAP sehingga proses bisnis dapat berjalan lebih efektif.

**BAB I**

# PROSES DESAIN DAN IMPLEMENTASI

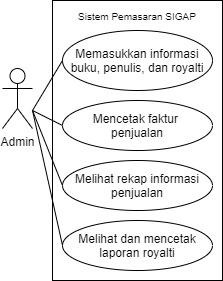
UGM Press sudah mempunyai sistem informasi untuk membantu proses bisnis mereka yang bernama Sistem Informasi Gadjah Mada Press (SIGAP). Sistem informasi ini mampu mengotomatisasi proses bisnis manual pada tahap pra-produksi (pembuatan naskah, *review*, editorial, desain *cover*, dan *layout*) dan produksi (menerima order cetak, melakukan percetakan, konfirmasi percetakan selesai). Namun pada bagian pasca-produksi (sistem pemasaran) proses bisnis masih dilakukan secara konvensional.

UGM Press memiliki alur bisnis berupa tahap-tahap dalam melaksanakan operasionalnya. Alur ini diawali dengan penulis menyerahkan naskah yang akan diterbitkan. Naskah kemudian diulas oleh bagian redaksi. Jika bagian redaksi menyetujui, maka naskah akan dikirimkan ke bagian penerbitan. Di bagian penerbitan naskah melewati proses *layouting.* Setelah tahap ini selesai maka UGM Press akan meminta persetujuan pengarang untuk memulai proses pencetakan. Penulis akan mendapatkan royalti dari hasil penjualan buku setiap 6 bulan sekali.

Alur proses pemasaran dan penghitungan royalti dimulai saat ada buku yang terjual dan teridentifikasi oleh sistem. Faktur akan dikeluarkan oleh sistem ketika terdapat buku yang terjual, yang kemudian faktur tersebut akan dikirimkan kepada bagian gudang agar dilakukan *update* stok buku. Faktur tersebut juga sebagai bukti untuk melakukan proses pencairan royalti kepada *stakeholder* yang berkaitan. Kemudian staf akan membuat laporan pembagian atau perhitungan royalti menggunakan data yang diperoleh.

Akan tetapi, untuk proses penghitungan royalti di UGM Press masih dilakukan secara konvensional. Staf UGM Press akan menghitung royalti dari data penjualan dengan aplikasi Microsoft Excel.

Sebagai gambaran hubungan fungsi sistem dengan pengguna, dapat dijelaskan oleh *use case diagram* sebagai berikut :



Gambar Use case diagram Sistem Royalti SIGAP

Untuk langkah pertama, admin sistem perlu memasukkan informasi yang diperlukan untuk melacak penjualan sebuah buku, yang meliputi nama buku, penulis, dan informasi royalti yang telah disepakati dengan penulis. Setiap kali penjualan berhasil dilakukan, maka sistem akan mencetak faktur dan memperbarui data di sistem. Kemudian admin dapat melihat informasi terbaru dari data penjualan sebuah buku. Admin juga dapat melihat rekap laporan penjualan, atau draft laporan hingga waktu tertentu. Secara periodis, admin perlu mencetak laporan royalti kepada penulis.

Sistem SIGAP yang kami rancang merupakan sistem untuk bagian pemasaran UGM Press, sehingga kami memiliki batasan permasalahan sebagai berikut :

1. Pengguna sistem adalah admin SIGAP. Pengguna memiliki akses penuh atas semua proses yang dijalankan sistem.
2. Proses yang dijalankan sistem yaitu penampilan data yang berkaitan dengan royalti, rekap informasi penjualan lampau, dan pembuatan faktur berkaitan dengan penjualan buku.
3. Data royalti penjualan dapat ditampilkan secara keseluruhan maupun secara perorangan. Sistem menampilkan data dalam bentuk tabel yang berisi nama penulis, judul buku, dan royalti yang didapatkan.
4. Sistem pemasaran dapat membuat faktur transaksi untuk setiap penulis. Faktur terbatas dalam satu periode royalti dan berbentuk tabel yang berisi nama penulis, judul buku, jumlah buku yang terjual, persentase royalti, dan nilai royalti yang akan diterima oleh penulis.

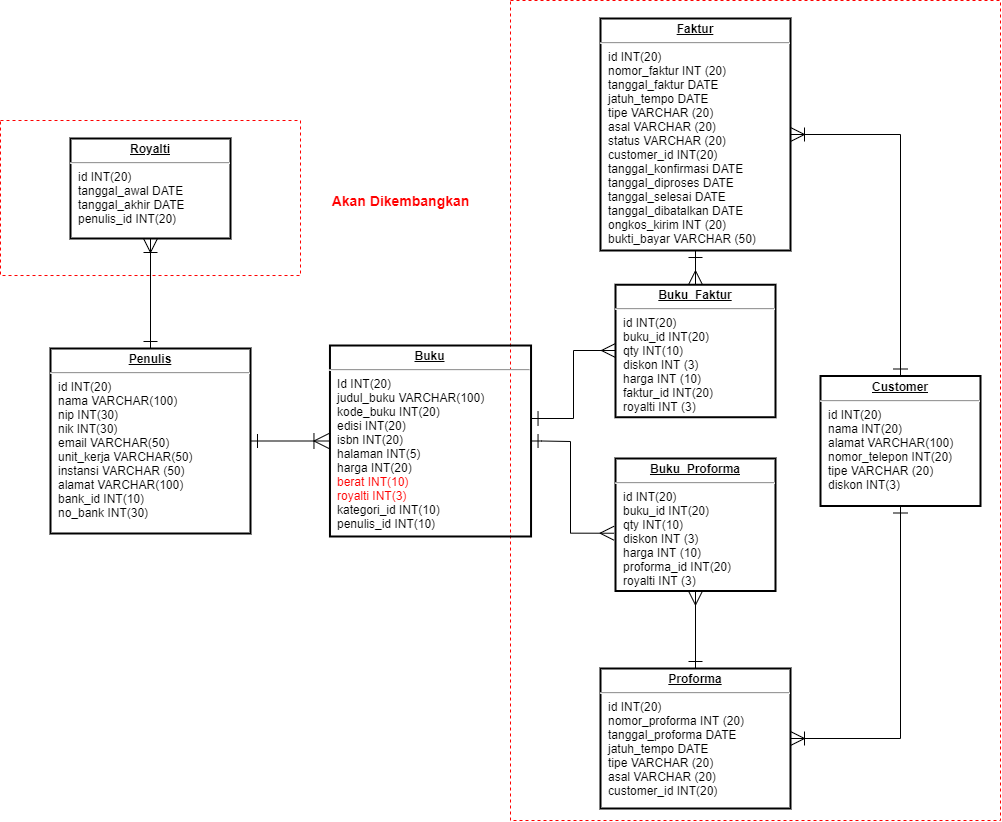
## Proses Desain Back-End

SIGAP merupakan sistem informasi yang berbasis *website* dengan arsitektur *Model View Controller* (MVC). Arsitektur MVC merupakan salah satu arsitektur yang paling umum dalam pengembangan aplikasi berbasis *website*. Arsitektur MVC membagi komponen menjadi tiga bagian, yaitu *model*, *view*, dan *controller*.

Pengembangan *back*-*end* SIGAP menggunakan bahasa PHP dengan *framework* Codeigniter. *Database* yang digunakan adalah *database* MySQL. Pengembangan menggunakan model arsitektur *Model View Controller* (MVC).

SIGAP merupakan sistem informasi yang berbasis *website* sehingga pertukaran data dilakukan dengan protokol HTTPS. Format data yang digunakan adalah JSON (JavaScript *Object Notation*).

Pengembangan dimulai dengan perancangan *Entity Relationship Diagram* (ERD) sebagai dasar rancangan *database*. Rancangan *database* dibagi menjadi tiga bagian utama yaitu Faktur, Royalti, dan Proforma. Bagian faktur berfungsi untuk mencatat data transaksi penjualan. Bagian royalti mencatat data periode royalti untuk tiap penulis. Sedangkan bagian proforma berfungsi mencatat simulasi pembelian atau *draft* faktur yang kemudian dapat diubah menjadi faktur. Gambar 2 menunjukkan ERD untuk rancangan pengembangan SIGAP.



Gambar ERD pengembangan SIGAP

Terdapat 8 tabel yang merupakan rancangan basis data yang akan digunakan pada pengembangan sistem SIGAP. Berikut merupakan tabel beserta atribut dan penjelasannya.

1. Tabel *customer*

Tabel *customer* berfungsi untuk menyimpan identitas *customer* dan dapat digunakan sebagai keperluan pembuatan faktur atau proforma. Tabel ini memiliki 6 kolom yang terdiri sebagai berikut.

Tabel *Data Dictionary* Tabel *customer*

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Size* | *Description* |
| *customer\_id* | int | 10 | *Unique id* untuk setiap *customer* |
| *name* | varchar | 50 | Nama dari setiap *customer* |
| *address* | varchar | 100 | Alamat dari setiap *customer* |
| *phone\_number* | varchar | 20 | Nomor telepon dari setiap *customer* |
| *type* | varchar | 25 | Jenis masing-masing *customer* (*types of membership*) |
| *email* | varchar | 50 | *Email* dari setiap *customer* |

1. Tabel *discount*

Tabel *discount* berfungsi untuk menyimpan besarnya diskon untuk masing-masing jenis *customer*. Hal ini diperlukan agar ketika pengguna membuat faktur dan memasukkan identitas *customer*, diskon akan terisi secara otomatis pada kolom diskon masing-masing buku sesuai dengan jenis *customer*. Tabel ini terdiri dari 3 kolom yang terdiri sebagai berikut.

Tabel *Data Dictionary* Tabel *discount*

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Size* | *Description* |
| *id* | int | 3 | *Unique id* untuk masing-masing diskon berdasarkan jenis *customer* |
| *membership* | varchar | 50 | Jenis *customer* untuk membedakan diskonnya masing-masing |
| *discount* | int | 3 | Diskon yang diberikan terhadap jenis *customer* tertentu |

1. Tabel *invoice*

Tabel *invoice* berfungsi untuk menyimpan detail suatu faktur. Faktur disimpan guna keperluan rekap data dan sebagai data acuan untuk menentukan besarnya royalti setiap penulis, karena besar kecilnya jumlah royalti ditentukan dari besarnya jumlah buku yang terjual. Data masing-masing faktur juga untuk diserahkan kepada *customer* sebagai bukti pemesanan buku. Tabel ini terdiri dari 19 kolom yang terdiri sebagai berikut.

Tabel *Data Dictionary* Tabel *invoice*

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Size* | *Description* |
| *invoice\_id* | *int* | 10 | *Unique id* untuk masing-masing *invoice* |
| *number* | *varchar* | 50 | Nomor *invoice* / faktur |
| *type* | *varchar* | 25 | Jenis faktur untuk masing-masing *invoice* |
| *source* | *varchar* | 100 | Tempat asal buku diambil untuk pesanan masing-masing faktur |
| *due\_date* | *datetime* |  | Tanggal jatuh tempo *invoice* / faktur |
| *customer\_id* | *varchar* | 20 | *Unique id* untuk setiap *customer* sebagai *foreign key* |
| *status* | *varchar* | 25 | Status untuk masing-masing *invoice* / faktur |
| *issued\_date* | *timestamp* |  | Tanggal dibuatnya masing-masing *invoice* / faktur |
| *confirm\_date* | *timestamp* |  | Tanggal konfirmasi setiap *invoice* / faktur |
| *preparing\_deadline* | *timestamp* |  | Tanggal jatuh tempo untuk persiapan pesanan masing-masing *invoice* / faktur |
| *preparing\_start\_date* | *timestamp* |  | Tanggal mulainya persiapan pesanan masing-masing *invoice* / faktur |
| *preparing\_end\_date* | *timestamp* |  | Tanggal berakhirnya persiapan pesanan masing-masing *invoice* / faktur |
| *finish\_date* | *timestamp* |  | Tanggal pesanan *invoice* / faktur selesai diproses |
| *cancel\_date* | *timestamp* |  | Tanggal dibatalkannya masing-masing *invoice* / faktur |
| *preparing\_staff* | *varchar* | 100 |  |
| *source\_library\_id* | *varchar* | 20 |  |
| *delivery\_fee* | *int* | 20 | Harga ongkos kirim masing-masing invoice / faktur |
| *total\_weight* | *int* | 20 | Total berat pesanan masing-masing invoice / faktur |
| *receipt* | *varchar* | 60 | Bukti bayar setiap *invoice* / faktur |

1. Tabel *invoice\_book*

Tabel *invoice\_book* berfungsi untuk menyimpan buku yang dipesan pada setiap faktur. Tabel ini terdiri dari 7 kolom sebagai berikut.

Tabel *Data Dictionary* Tabel *invoice\_book*

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Size* | *Description* |
| *invoice\_book\_id* | int | 10 | *Unique id* untuk masing-masing tabel pesanan buku pada setiap *invoice* / faktur |
| *invoice\_id* | varchar | 20 | *Unique id* untuk masing-masing invoice sebagai *foreign key* |
| *book\_id* | varchar | 20 | *Unique id* untuk setiap buku sebagai *foreign key* |
| *qty* | int | 10 | Jumlah masing-masing buku yang dipesan |
| *discount* | int | 3 | Besarnya diskon yang diberikan untuk setiap buku yang dipesan |
| *price* | int | 10 | Harga untuk masing-masing buku |
| *royalty* | int | 3 |  |

1. Tabel proforma

Tabel proforma berfungsi untuk menyimpan detail suatu proforma. Proforma adalah suatu penawaran kepada *customer* yang berisikan buku yang diinginkan atau ditawarkan kepada *customer*. Tabel ini terdiri dari 6 kolom yang terdiri sebagai berikut.

Tabel *Data Dictionary* Tabel proforma

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Size* | *Description* |
| proforma\_id | int | 10 | *Unique id* untuk setiap proforma yang dibuat |
| *number* | varchar | 50 | Nomor proforma |
| *source* | varchar | 100 |  |
| *due\_date* | datetime |  | Tanggal jatuh tempo setiap proforma |
| *customer\_id* | varchar | 20 | *Unique id* untuk setiap customer sebagai *foreign key* |
| *issued\_date* | datetime |  | Tanggal dibuatnya setiap proforma |

1. Tabel proforma\_*book*

Tabel proforma\_*book* berfungsi untuk menyimpan buku yang dipesan pada setiap proforma. Tabel ini terdiri dari 6 kolom yang terdiri sebagai berikut.

Tabel *Data Dictionary* Tabel proforma\_*book*

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Size* | *Description* |
| *id* | int | 10 | *Unique id* untuk masing-masing tabel pesanan buku pada setiap proforma |
| *proforma\_id* | varchar | 20 | *Unique id* untuk setiap proforma sebagai *foreign key* |
| *book\_id* | varchar | 20 | *Unique id* untuk setiap buku sebagai *foreign key* |
| *qty* | int | 10 | Jumlah masing-masing buku yang dipesan |
| *discount* | int | 3 | Besarnya diskon yang diberikan untuk setiap buku yang dipesan |
| *price* | int | 10 | Harga untuk masing-masing buku |

1. Tabel *royalty*

Tabel *royalty* berfungsi untuk menyimpan pengajuan dan pembayaran royalti setiap penulis pada periode yang ditentukan. Pengguna dapat menentukan tanggal awal dan akhir periode pengajuan royalti yang akan dibayar sehingga tidak perlu menunggu setiap 6 bulan sekali seperti proses bisnis pemasaran UGM Press sekarang. Tabel ini terdiri dari 7 kolom yang terdiri sebagai berikut.

Tabel *Data Dictionary* Tabel *royalty*

|  |  |  |  |
| --- | --- | --- | --- |
| *Field Name* | *Data Type* | *Size* | *Description* |
| *royalty\_id* | int | 10 | *Unique id* untuk setiap royalti yang diajukan |
| *author\_id* | varchar | 20 | *Unique id* untuk setiap author / penulis sebagai *foreign key* |
| *start\_date* | datetime |  | Tanggal mulai periode royalti yang akan diajukan |
| *end\_date* | datetime |  | Tanggal akhir periode royalti yang akan diajukan |
| *status* | varchar | 20 | Status royalti setiap penulis pada periode tertentu |
| *paid\_date* | datetime |  | Tanggal pembayaran royalti pada setiap pengajuan royalti |
| *receipt* | varchar | 200 | Bukti pembayaran royalti setiap penulis |

1. Tabel *royalty\_payment*

Tabel *Data Dictionary* Tabel *royalty\_payment*

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Size | Description |
| id | int | 10 |  |
| author\_id | varchar | 50 |  |
| last\_paid\_date | datetime |  |  |
| last\_request\_date | datetime |  |  |
| status | varchar | 20 |  |

## Proses Desain *Front*-*End*

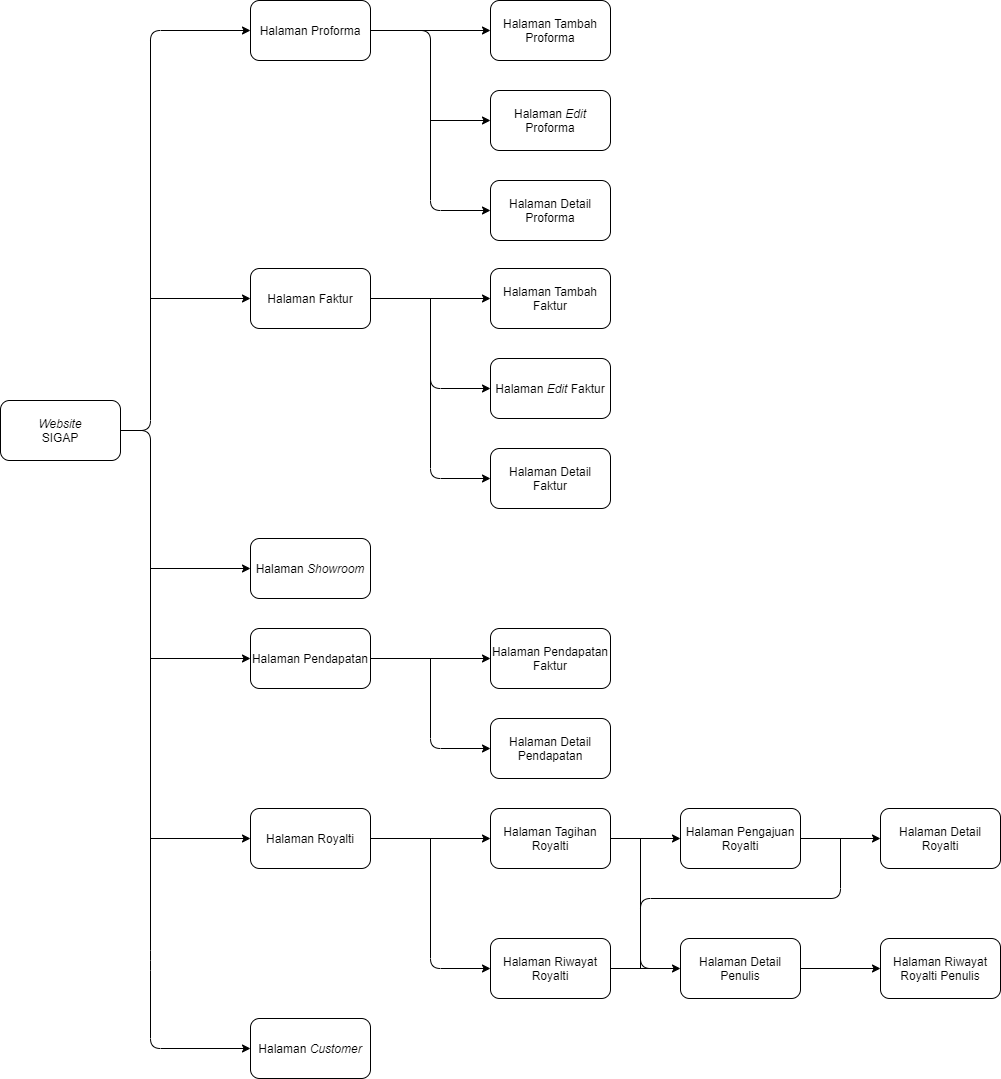
Pengembangan front-end pada SIGAP menggunakan HTML dan CSS dengan *framework* Bootstrap. Pada arsitektur MVC, pengembangan front-end dilakukan di bagian *view.* Komponen *website* yang akan ditampilkan ke pengguna dibuat di *view* yang kemudian akan ditampilkan oleh *controller* sesuai dengan pengaturan *route*.

Komponen *website* yang ditampilkan pada setiap halaman, seperti *sidebar* dipisahkan pada *file view* sendiri yang kemudian akan dipanggil di setiap *file view* lainnya. Metode ini membantu memudahkan pengembangan sehingga *source code* dapat lebih mudah dibaca. Selain itu, metode ini juga memudahkan pengembang saat akan melakukan perubahan pada *sidebar* karena tidak perlu mengubah satu per satu di setiap *view* halaman.

Sedangkan untuk komponen *view* yang perlu mengirimkan/menerima data secara dinamis, misalnya form penambahan data faktur, menggunakan *Asynchronous* JavaScript And XML (AJAX). AJAX digunakan untuk mengirim data ke *controller* menerima data dari *controller* tanpa perlu melakukan *refresh* halaman. Penggunaan AJAX membantu menciptakan halaman *website* yang dinamis.

### Struktur Sistem Informasi

Untuk mempermudah proses implementasi sistem informasi, berikut merupakan rancangan pengembangan struktur sistem informasi pemasaran SIGAP berdasarkan fungsi-fungsi untuk mengatasi permasalahan yang sudah didefinisikan.



Gambar Rancangan Struktur Pengembangan Sistem Informasi Pemasaran SIGAP

Rancangan sistem informasi pemasaran SIGAP dibagi menjadi beberapa halaman sesuai dengan fungsinya masing-masing sebagai berikut.

1. Halaman Proforma

Merupakan halaman yang memuat informasi keseluruhan proforma yang sudah dibuat. Melalui halaman ini, pengguna dapat menuju ke halaman tambah proforma, edit proforma, dan detail proforma untuk melakukan aksi tertentu sesuai kebutuhan.

1. Halaman Tambah Proforma

Memungkinkan pengguna untuk menambah proforma untuk keperluan pemasaran dan kolom data yang disediakan merupakan informasi yang dibutuhkan pemasaran untuk membuat suatu proforma.

1. Halaman *Edit* Proforma

Pengguna dapat melakukan perubahan data terhadap suatu informasi proforma tertentu jika terdapat kesalahan melalui halaman ini. Untuk menuju ke halaman ini, akan disediakan tombol edit sehingga memudahkan pengguna untuk menemukan dan menuju ke halaman ini.

1. Halaman Detail Proforma

Halaman yang memuat seluruh informasi suatu proforma tertentu. Melalui halaman ini pengguna dapat melihat informasi proforma tersebut secara rinci dan fitur unduh proforma akan ditempatkan di halaman ini.

1. Halaman Faktur

Halaman yang memuat data keseluruhan jenis faktur yang dibuat. Masing-masing faktur akan ditampilkan statusnya sehingga bagian pemasaran UGM Press mudah mencari dan memproses pesanan sesuai faktur. Melalui halaman ini, pengguna dapat menuju ke halaman tambah faktur, *edit* faktur, dan detail faktur untuk melakukan aksi tertentu sesuai kebutuhan.

1. Halaman Tambah Faktur

Pengguna dapat melakukan aksi tambah faktur dengan mengisi data yang dibutuhkan sesuai kolomnya masing-masing. Faktur yang dapat dibuat melalui halaman ini adalah faktur dengan jenis Kredit, Tunai, dan *Online*. Faktur dengan jenis *Showroom* dibuat terpisah disesuaikan dengan permintaan dari pemasaran UGM Press.

1. Halaman *Edit* Faktur

Pengguna dapat melakukan perubahan data faktur jika faktur tersebut belum dikonfirmasi oleh bagian pemasaran UGM Press. Data faktur nantinya akan ditampilkan sesuai dengan kolomnya masing-masing dan pengguna dapat merubah data tersebut. Untuk menuju halaman ini, akan disediakan tombol *edit* pada faktur yang memenuhi syarat.

1. Halaman Detail Faktur

Memuat seluruh informasi faktur tertentu seperti identitas *customer*, detail status faktur, dan tabel buku yang dipesan. Fungsi unduh faktur juga akan ditempatkan pada halaman ini dan akan muncul jika faktur tersebut sudah dikonfirmasi oleh bagian pemasaran UGM Press.

1. Halaman *Showroom*

Halaman ini dirancang untuk keperluan *showroom* yang membutuhkan kecepatan dalam pelayanan pelanggan. Pengguna memungkinkan untuk tidak memasukkan identitas *customer* dalam proses pembuatan faktur *showroom* dan akan dikategorikan sebagai *customer* umum.

1. Halaman Pendapatan

Halaman pendapatan akan dibagi menjadi 2 *tab* sebagai berikut.

1. Halaman Pendapatan Faktur

Menampilkan jumlah pendapatan seluruh jenis faktur dalam bentuk grafik sehingga pengguna dapat mengetahui jumlah pendapatan masing-masing jenis faktur setiap satu bulannya. Fungsi unduh laporan pendapatan akan ditempatkan pada halaman ini.

1. Halaman Detail Pendapatan

Guna kebutuhan dalam menentukan kebijakan oleh pihak pemasaran UGM Press seperti *marketing*, *sales promotion*, dll, halaman ini akan menampilkan perbandingan jumlah pendapatan seluruh jenis faktur dalam bentuk *pie chart* dalam periode tertentu.

1. Halaman Royalti

Halaman royalti akan dibagi menjadi 2 *tab*, yaitu halaman tagihan royalti dan halaman riwayat royalti.

1. Halaman Tagihan Royalti

Halaman ini akan menampilkan penulis dan jumlah royalti masing-masing penulis yang harus dibayarkan oleh UGM Press. Jika penulis tertentu tidak ada royalti yang harus dibayar oleh UGM Press, maka nama penulis tersebut tidak akan muncul di halaman ini. Jika akan melakukan pengajuan royalti, pengguna dapat menekan tombol pada halaman ini.

1. Halaman Riwayat Royalti

Halaman ini memuat riwayat royalti seluruh penulis, baik yang statusnya diajukan atau dibayar. Halaman ini akan memudahkan pengguna untuk melihat jumlah keseluruhan royalti yang didapatkan oleh masing-masing penulis,.

1. Halaman Pengajuan Royalti

Halaman ini berfungsi untuk mengajukan pembayaran royalti pada periode tertentu dan sifat periodenya adalah fleksibel, karena pengguna dapat memasukkan tanggal awal dan akhir periode yang mau diajukan. Halaman ini juga berfungsi untuk mengubah status royalti penulis menjadi sudah dibayar dengan memasukkan bukti bayar royalti.

1. Halaman Detail Royalti

Menampilkan detail royalti penulis tertentu pada suatu periode. Akan disediakan tabel yang berisikan informasi judul buku dan jumlah buku yang terjual sehingga pengguna mengetahui asal jumlah besaran royalti yang diterima penulis.

1. Halaman Detail Penulis

Halaman ini menampilkan identitas penulis. Halaman ini dapat diakses dengan cara menekan nama penulis tertentu pada halaman royalti atau bisa juga melalui halaman penulis.

1. Halaman Riwayat Royalti Penulis

Merupakan salah satu *tab* dalam halaman detail penulis yang berisikan riwayat royalti beserta waktu periodenya.

1. Halaman *Customer*

Halaman ini akan menampilkan identitas *customer* dan pengguna dapat juga menambah atau mengubah data *customer* melalui halaman ini. Fungsi tambah dan *edit* *customer* dirancang menggunakan *modal*. Modal untuk melakukan perubahan besar diskon untuk masing-masing jenis *customer* akan disediakan melalui halaman ini dengan cara menekan tombol diskon.

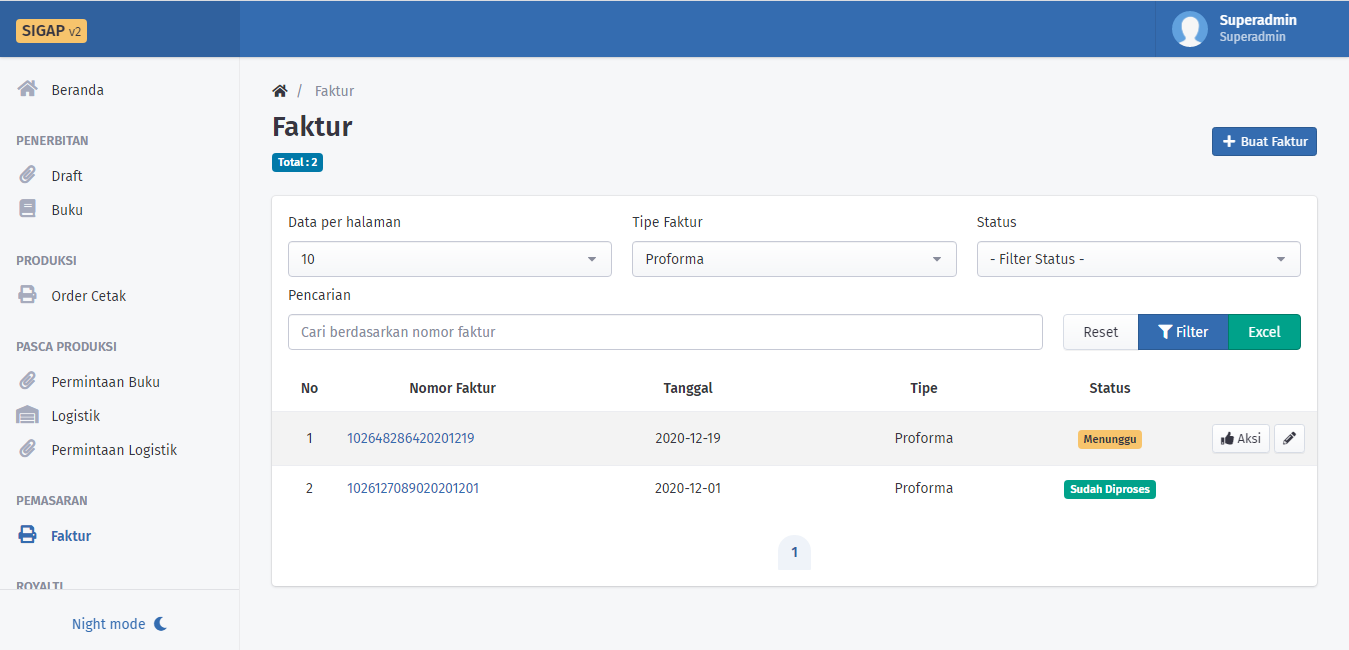
## Proses Desain User Experience

Pada tahapan desain *User Experience*, penulis merancang *High Fidelity* dan *prototype* sistem pemasaran SIGAP berdasarkan kebutuhan pengguna. Dalam proses perancangan, penulis berdiskusi dengan pengguna sistem terkait alur bisnis dan fungsionalitas apa saja yang dibutuhkan dalam sistem pemasaran SIGAP. Tampilan kemudian dirancang sesuai dengan alur fungsionalitas yang sesuai dengan alur bisnis pemasaran UGM Press berbasis *website*. Berikut merupakan rancangan desain tampilan sistem pemasaran SIGAP.



### Halaman Faktur

Setiap terjadi penjualan buku, admin sistem perlu mencetak faktur penjualan yang kemudian akan diberikan pada pengurus gudang UGM Press. Faktur tersebut juga akan digunakan sebagai verifikasi untuk memastikan transparansi pada keseluruhan sistem pemasaran. Pada gambar ,pengguna dapat melihat semua faktur yang pernah dibuat dan juga memuat informasi yang berkaitan, seperti nomor faktur, tanggal, tipe faktur, dan statusnya. Pengguna juga dapat melakukan pencarian spesifik dan melakukan penyaringan data.



Gambar  *Halaman “Faktur”*

Pengguna dapat menambahkan faktur baru dengan mengakses link dengan menekan tombol tambah faktur pada gambar . Setelah itu, pengguna akan dibawa pada halaman seperti pada gambar .Pengguna perlu menginput data yang berkaitan untuk menambahkan faktur tersebut.



Gambar  Halaman *“Pembuatan Faktur”*

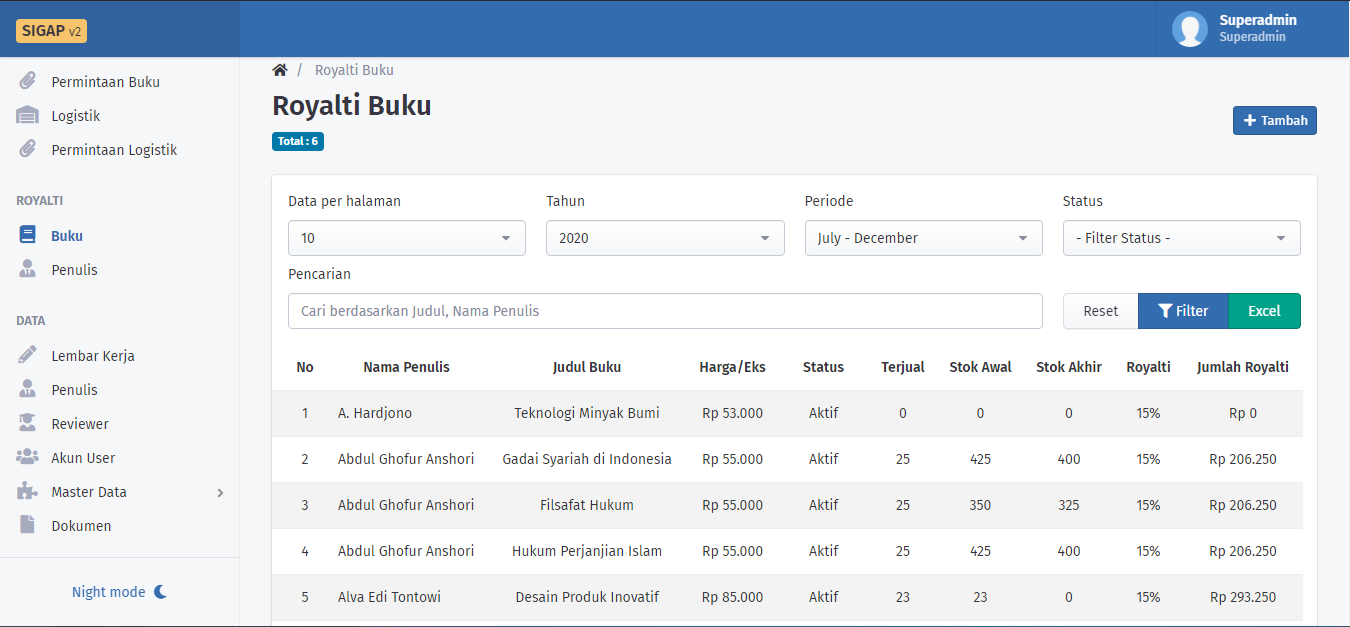
Setelah menambahkan faktur baru, pengguna dapat melihat detailnya dengan mengakses link dengan menekan nomor fakturnya pada gambar . Tampilannya seperti pada gambar , yang memuat informasi seperti judul buku, nama penulis, dan lainnya.



Gambar  Halaman *“Transaksi Buku”*

### Halaman Royalti

Admin sistem dapat mengakses dan melihat informasi utama yang perlu diketahui tentang royalti buku pada halaman “Royalti Buku”. Pada halaman ini, admin dapat melakukan filter data, mengurutkan berdasarkan suatu syarat tertentu, atau melakukan pencarian khusus untuk sebuah buku. Gambar memberikan gambaran pada rancangan tampilan halaman tersebut.



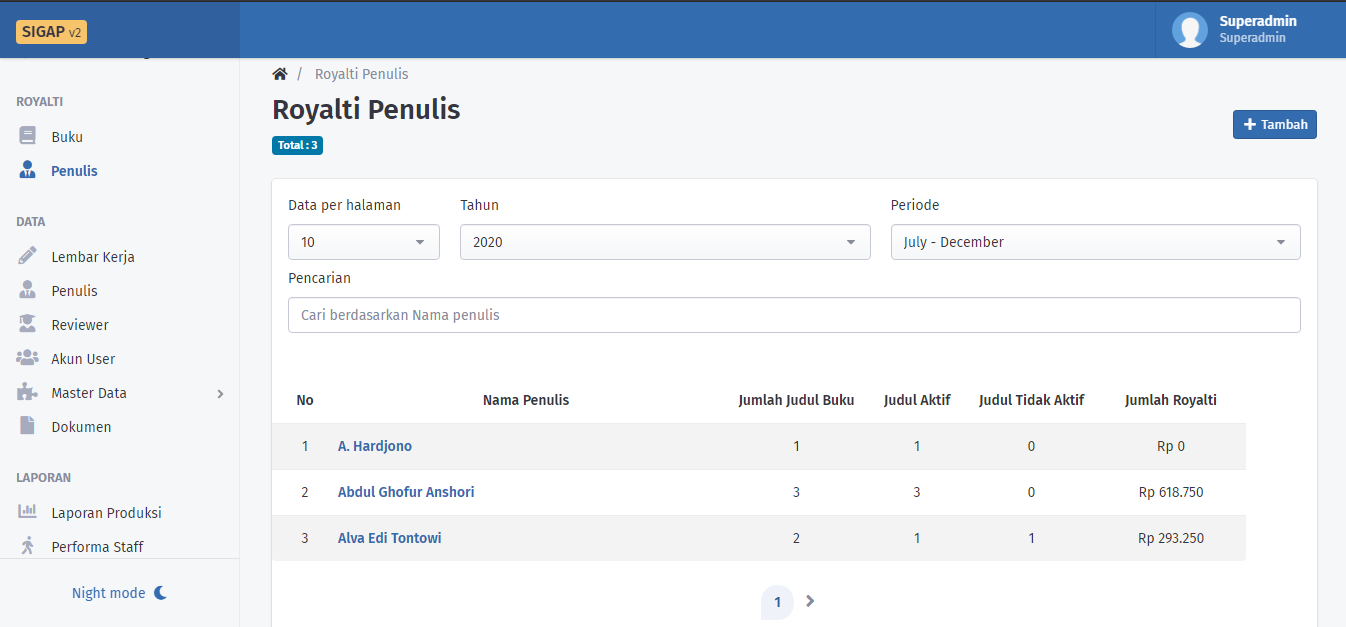
Gambar Halaman *“Royalti Buku”*

Secara umum, penulis akan menerima royalti sebanyak 15% apabila tidak ada syarat atau bantuan tambahan yang diterima pada saat kesepakatan awal dengan pihak UGM Press. Pada gambar 17, dapat dilihat bahwa terdapat beberapa buku dengan harga Rp 55.000 dan Rp 85.000. Pada pembelian buku dengan jumlah 0, tentunya penulis tidak akan menerima royalti apapun. Sedangkan pada penjualan lebih dari 1 buku, berikut ini adalah jumlah yang akan diterima oleh penulis:

1. Rp 55.000 \* 25 \* 15% = Rp 206.250
2. Rp 85.000 \* 23 \* 15% = Rp 293.250

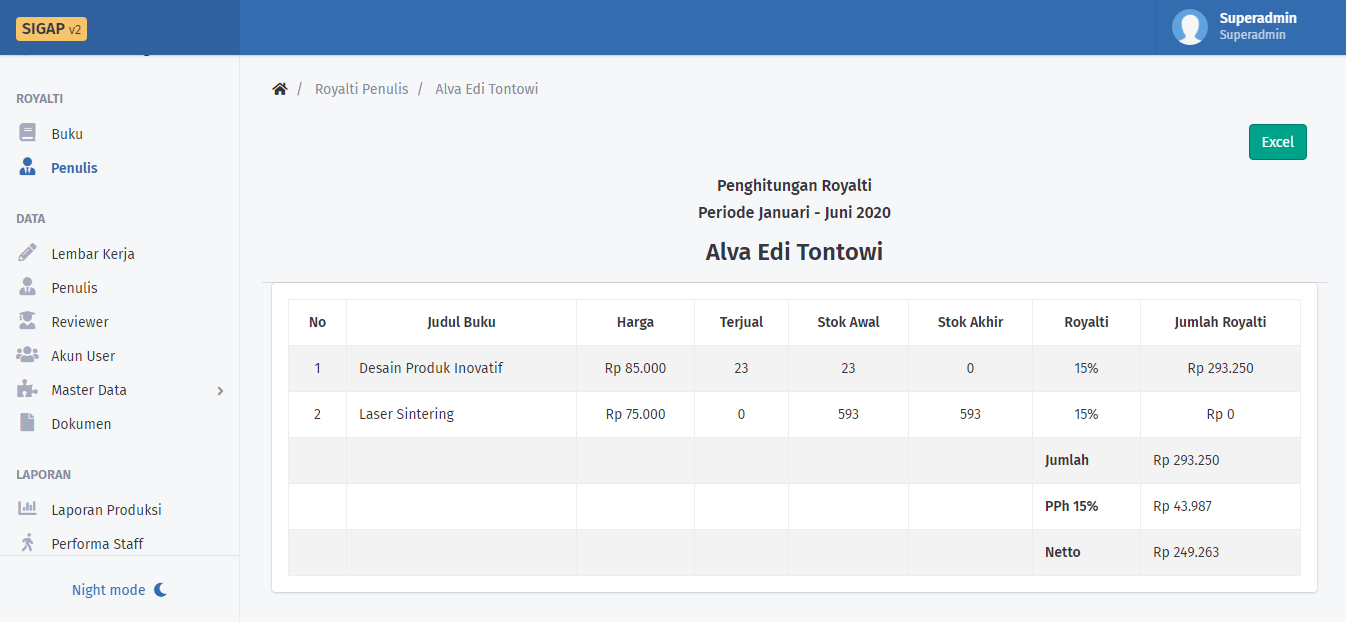
Dengan mengakses tombol “Tambah” yang terdapat di kanan atas halaman, admin dapat menambah daftar buku dengan informasinya pada halaman royalti ini. Admin sistem juga dapat membuat laporan royalti untuk seorang penulis.

Pada bagian navigasi, terdapat tombol untuk membuka halaman “Royalti Penulis”. Admin sistem dapat melihat daftar penulis dan informasi yang berkaitan dengan penulis, seperti jumlah judul buku, judul aktif, judul tidak aktif, dan jumlah royalti yang akan diterima pada periodenya melalui halaman ini. Gambar memberikan tampilan rancangan halaman tersebut.



Gambar  *Halaman “Royalti Penulis”*

Apabila admin sistem mengakses halaman penulis secara individu, maka admin dapat melihat informasi informasi royalti untuk setiap buku yang ditulis oleh penulis yang diakses halamannya. Admin juga dapat membuat laporan royalti untuk penulis melalui halaman ini. Gambar adalah rancangan dari halaman tersebut.



Gambar  *Halaman penulis*

## Penerapan Teknologi

Pengembangan SIGAP menggunakan framework dan beberapa library tambahan untuk mempermudah proses implementasi fitur.Beberapa framework dan library yang digunakan untuk implementasi fitur pada pengembangan aplikasi kelompok kami dapat dilihat pada tabel **xxx**.

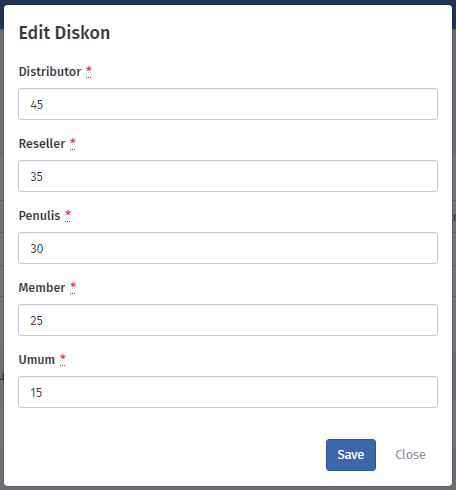
|  |  |  |
| --- | --- | --- |
| **No** | **Nama Framework/Library** | **Keterangan / Deskripsi** |
| 1 | CodeIgniter | Framework PHP yang menerapkan MVC. CodeIgniter memiliki struktur file yang mudah dipahami, syntax yang terstruktur, serta library yang mencukupi kebutuhan aplikasi kami. |
| 2 | Bootstrap | Framework *front-end* untuk mengembangkan tampilan yang sederhana dan responsif, serta penulisan syntax yang rapi. |
| 3 | Chart.js | Library yang ditulis dengan bahasa *javascript* untuk membuat grafik yang interaktif. Pada aplikasi kami, digunakan untuk membuat grafik pada modul pendapatan. |
| 4 | PHPSpreadsheet | Library yang dapat digunakan pada framework CI, untuk membaca atau membuat file *spreadsheet* dengan format seperti *Excel.* |
| 5 | MySQL | Teknologi database yang kami gunakan untuk menyimpan, memperbarui, atau mengambil data di aplikasi kami. |

## Implementasi Fitur Customer dan Diskon

Pada setiap penjualan buku, salah satu informasi yang perlu dicatat adalah berkaitan dengan informasi customernya. Pada umumnya, terdapat lima jenis tipe *customer* yang akan memperoleh nilai diskon yang berbeda-beda. Pembagiannya dapat dilihat pada tabel **xxx**.

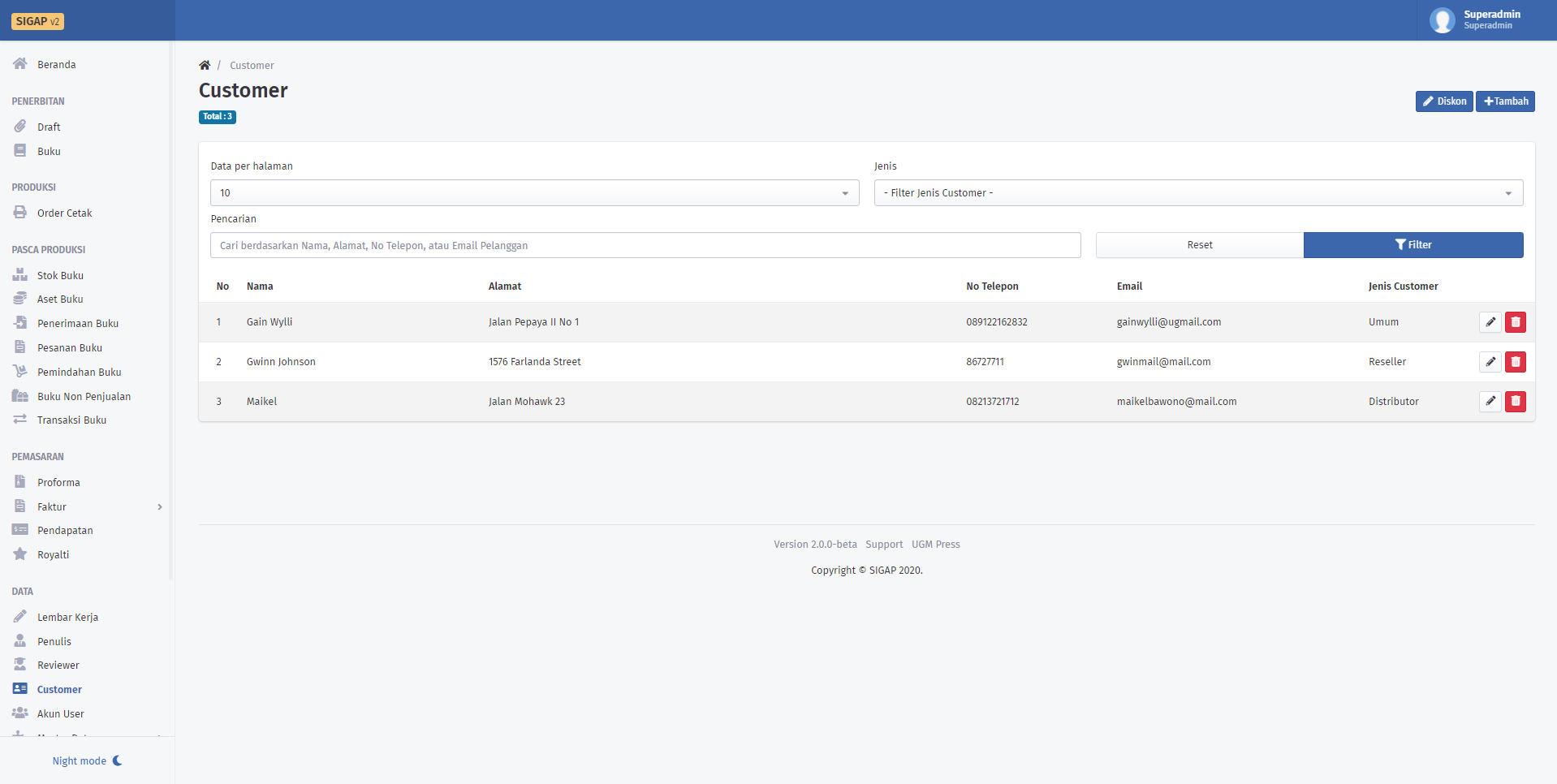
|  |  |  |
| --- | --- | --- |
| No | Jenis Customer | Diskon (%) |
| 1 | Distributor | 45 |
| 2 | Reseller | 35 |
| 3 | Penulis | 30 |
| 4 | Member | 25 |
| 5 | Umum | 15 |

Nilai diskon yang diperoleh untuk setiap jenis *customer* itu sendiri tidak pasti dan dapat berubah-ubah. Oleh karena itu, terdapat fitur yang memungkinkan admin sistem untuk mengubah nilai diskon yang diperoleh. Fitur tersebut berupa *modal* yang dapat diakses pada halaman utama *customer*. Gambar 10adalah tampilan fitur untuk mengubah nilai diskon untuk tiap jenis *customer*.



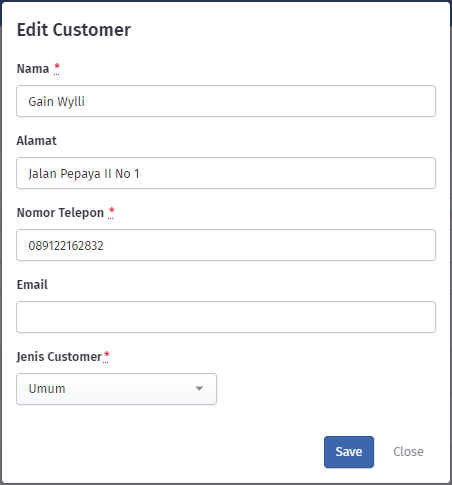
Gambar *Modal* pengaturan diskon customer

Pada halaman utama *customer* sendiri, terdapat fitur filter sama seperti halaman utama untuk fitur lainnya. Filter yang tersedia yaitu mencari menggunakan *dropdown* jenis *customer* atau menggunakan *keyword* berdasarkan nama, alamat, nomor telepon, atau email *customer.* Untuk mengubah informasi pada sebuah *customer*.



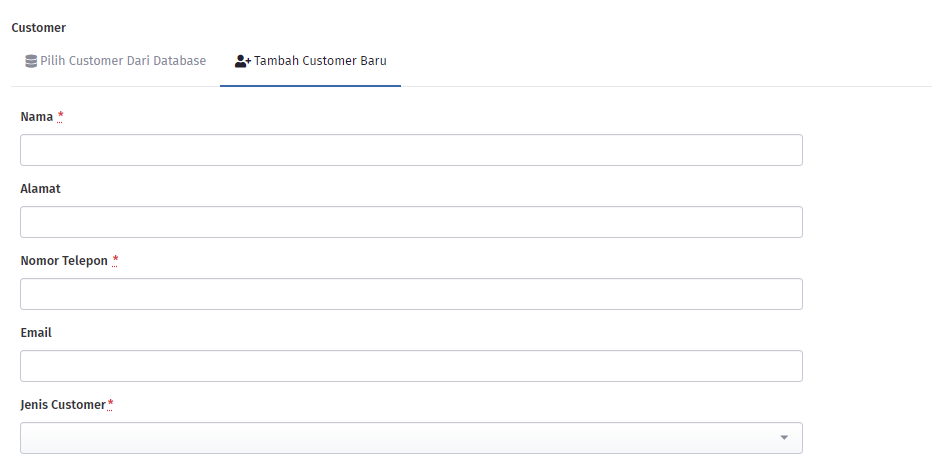
Gambar Halaman “*Customer*”

Pada halaman utama *customer*, dapat dilihat bahwa terdapat informasi mengenai nama, alamat, nomor telepon, email, serta jenis *customer*. Pada bagian kanan tabel, terdapat kolom untuk mengubah informasi setiap *customer*, dan tombol lainnya digunakan untuk menghapus *customer* dari *database*. Setelah menekan tombol *edit,* maka akan muncul tampilan *modal* untuk mengubah informasi yang sekarang telah ada. Berikut ini adalah tampilan *modal* tersebut.



Gambar *Modal* *edit customer*

Untuk menambahkan *customer* baru, pengguna dapat melakukannya dengan dua cara, yaitu dengan mengakses *modal* yang muncul setelah menekan tombol “Tambah” di kanan atas halaman utama *customer*, atau ketika menambahkan faktur baru. Gambar 13 menunjukkan tampilan *form* untuk menambah *customer* saat menambahkan faktur baru.



Gambar 13 *form* penambahan *customer* baru saat akan menambahkan faktur baru

Fitur *customer* dan diskon diatur melalui file *controller* bernama *Customer.php* dan *Customer\_model*.php yang terletak pada modul *customer*. Tabel **xxx** mendeskripsikan fungsi pada *controller*, sedangkan tabel **xxx** mendeskripsikan fungsi pada *model*.

|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | \_\_construct() | Memuat model dan helper yang akan digunakan, yaitu *customer\_model* serta *sales\_helper*. |
| 2 | index() | Mengarahkan pengguna ke halaman utama customer, dan mengirimkan data customer dan diskon dari customer\_model ke view. |
| 3 | api\_customer\_info() | Fungsi API yang digunakan untuk mengirimkan informasi seorang customer ketika modal edit customer ditekan. |
| 4 | add() | Menyimpan data customer baru ke dalam database. |
| 5 | edit() | Menyimpan perubahan data customer ke dalam database. |
| 6 | delete() | Menghapus seorang customer dari database. |
| 7 | edit\_discount() | Menyimpan perubahan diskon yang diperoleh dari jenis customer. |

|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | validate\_modal\_add() | Memvalidasi data *post* ketika akan menambahkan *customer* ke database. |
| 2 | validate\_modal\_edit() | Memvalidasi data *post* ketika akan memperbarui data *customer* ke database. |
| 3 | validate\_edit\_discount() | Memvalidasi input *post* ketika akan menambahkan data diskon yang diperoleh untuk setiap jenis *customer*. |
| 4 | get\_discount() | Mengambil nilai diskon yang dimiliki untuk setiap jenis *customer* dari database. |
| 5 | filter\_data() | Mengambil seluruh data *customer* dan total *record* yang ada dari database berdasarkan filter yang dipakai. |
| 6 | when() | Fungsi untuk membantu *query when()* di MySQL. |

## Implementasi Fitur Faktur

Faktur merupakan salah satu komponen inti dalam proses bisnis UGM Press. Faktur berfungsi untuk mencatat data penjualan buku di UGM Press. Faktur juga menjadi acuan dalam pembayaran royalti penulis.

UGM Press membagi faktur menjadi 4 jenis, yaitu faktur tunai, faktur *showroom*, faktur *online*, dan faktur kredit. Faktur tunai adalah faktur penjualan buku yang dibayar secara langsung. Faktur showroom adalah faktur untuk penjualan buku di *showroom* UGM Press. Faktur *online* adalah faktur penjualan buku yang dilakukan melalui *marketplace*. Faktur kredit adalah faktur penjualan buku yang pembayarannya dilakukan secara kredit.

Fitur faktur berfungsi untuk mencatat transaksi penjualan buku. Fitur ini menggunakan dua tabel pada database yaitu tabel faktur dan tabel buku faktur. Tabel faktur mencatat informasi umum mengenai faktur, seperti data customer, tanggal pembuatan, dan status faktur. Sedangkan tabel buku faktur berfungsi untuk mencatat data buku yang terjual, seperti judul buku, jumlah buku, harga buku, diskon, dan persentase royalti penulis.

Data faktur ditambahkan melalui halaman tambah faktur. Halaman ini berisi form untuk mengisi data faktur. Data *customer* dan buku yang akan ditambahkan berupa *dropdown* yang dilengkapi dengan *search bar* untuk memudahkan pencarian data. Data buku yang ditampilkan hanya buku yang data stoknya lebih dari nol sehingga tidak mungkin ada data faktur yang bukunya tidak tersedia.

Salah satu komponen data faktur adalah nomor faktur. Nomor faktur digunakan oleh pengguna untuk identifikasi faktur. Nomor faktur akan dibuat secara otomatis berdasarkan tanggal faktur dan jenis faktur.

Fitur faktur juga dilengkapi dengan pembuatan file *excel* dan PDF. Pembuatan file excel menggunakan *library* PHPOffice. Sedangkan untuk pembuatan file PDF digunakan *library* dompdf. Pembuatan file *excel* dan PDF digunakan oleh pengguna untuk proses rekap data faktur.

Fitur *customer* diatur melalui file *controller* bernama *Invoice.php* dan *Invoice\_model*.php yang terletak pada modul *invoice*. Tabel **xxx** mendeskripsikan fungsi pada *controller*, sedangkan tabel **xxx** mendeskripsikan fungsi pada *model*.

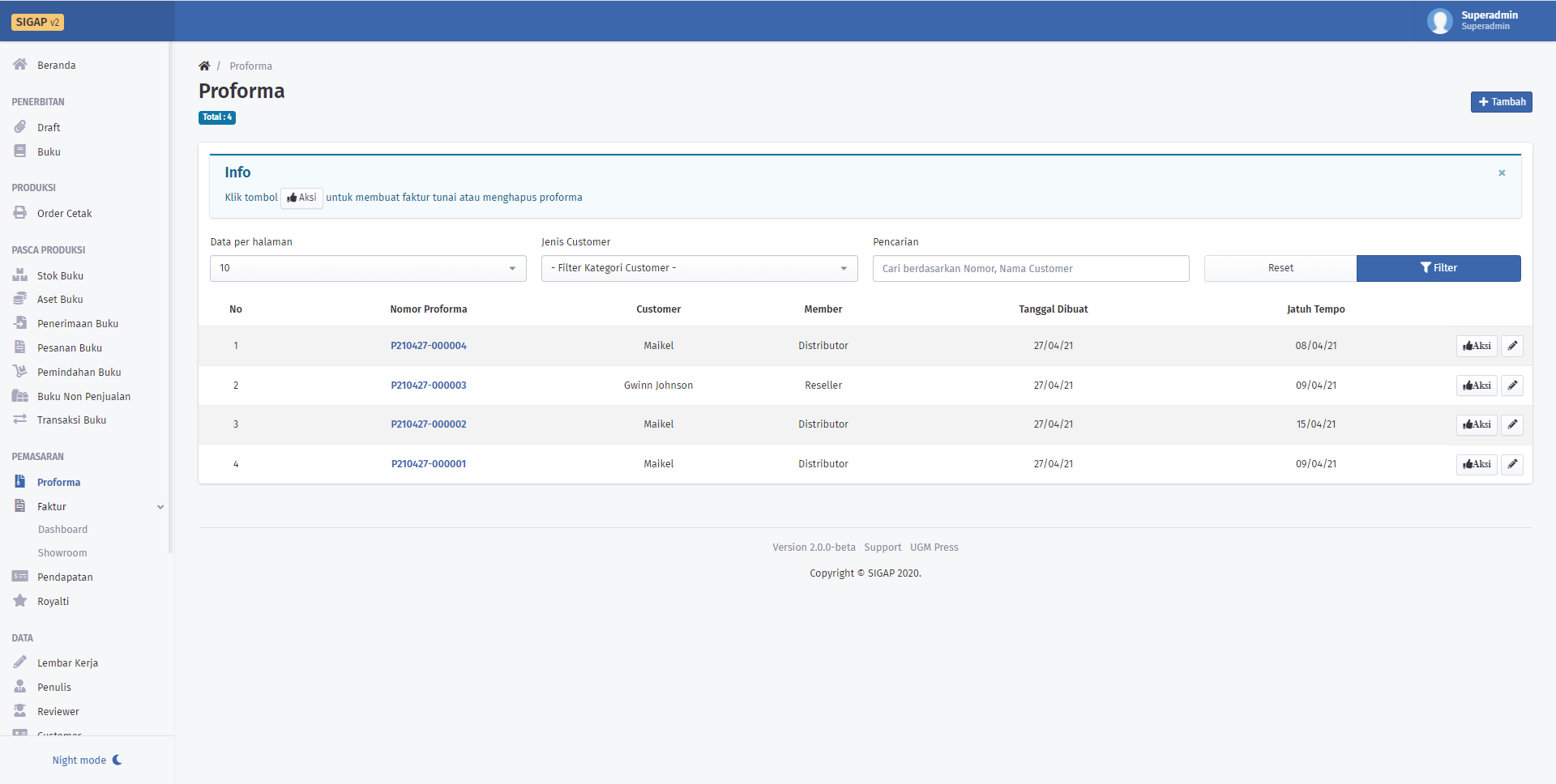
|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | \_\_construct() | Memuat model dan helper yang akan digunakan, yaitu *invoice*\_*model* serta *sales\_helper*. |
| 2 | index() | Mengarahkan pengguna ke halaman utama *invoice*, dan mengirimkan data *invoice* dari *invoice\_model* ke view. |
| 3 | view() | Mengarahkan pengguna ke halaman sebuah *invoice* dan mengirimkan data *invoice* tersebut ke view. |
| 4 | add() | Mengarahkan pengguna ke halaman untuk menambahkan *invoice* baru dan mengirimkan data *post*  untuk menambahkan *invoice* baru ke dalam database. |
| 5 | add\_showroom() | Mengarahkan pengguna ke halaman untuk menambahkan *invoice* baru berjenis *showroom*. |
| 6 | edit() | Mengarahkan pengguna ke halaman untuk mengubah data *invoice* sebelumnya dan mengirimkan data *post*  untuk memperbarui data *invoice*  ke dalam database. |
| 7 | action() | Memperbarui informasi status *invoice* ke tahap berikutnya, dan mengirimkan data *toast* ke view. |
| 8 | update\_delivery\_fee() | Memperbarui informasi ongkir untuk mengirim barang. |
| 9 | generate\_pdf() | Membuat file PDF untuk *invoice* yang jenisnya selain *showroom.* |
| 10 | showroom\_pdf() | Membuat file PDF untuk *invoice* yang jenisnya *showroom.* |
| 11 | generate\_excel() | Membuat file excel dengan mengambil data tabel dari index(). |
| 12 | api\_get\_book() | Fungsi API yang digunakan untuk mengirimkan informasi sebuah buku ketika pengguna memilih sebuah buku dari dropdown buku di *view* add\_showroom. |
| 13 | api\_get\_book\_dynamic\_stock() | Fungsi API yang digunakan untuk mengirimkan informasi sebuah buku ketika pengguna memilih sebuah buku dari dropdown buku di *view* add\_invoice. |
| 14 | api\_get\_customer() | Fungsi API yang digunakan untuk mengirimkan informasi sebuah *customer* ketika pengguna memilih seorang *customer* dari dropdown *customer* di *view* add\_invoice dan add\_showroom. |
| 15 | api\_get\_discount() | Fungsi API untuk mengirim informasi diskon dari jenis *customer* yang dipilih pada dropdown *customer* di *view* add\_invoice dan add\_showroom. |
| 16 | api\_get\_book\_dropdown() | Fungsi API untuk menambahkan data buku yang dihapus dari list kembali ke dropdown buku di *view* add\_invoice dan add\_showroom. |

|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | validate\_invoice() | Memvalidasi data *post* ketika akan menambahkan atau memperbarui *invoice* ke database. |
| 2 | fetch\_invoice\_id() | Mengambil data sebuah *invoice* dari database. |
| 3 | fetch\_invoice\_book() | Mengambil data *invoice* dan *invoice\_book* yang berkaitan dengan *invoice\_id* yang sama dari database. |
| 4 | fetch\_warehouse\_stock() | Mengambil data stok sebuah buku yang berada di *warehouse.* |
| 5 | fetch\_showroom\_stock() | Mengambil data stok sebuah buku yang berada di *showroom.* |
| 6 | fetch\_library\_stock() | Mengambil data stok sebuah buku yang berada di *library* tertentu*.* |
| 7 | get\_available\_book\_list() | Mengirimkan data dari fungsi fetch\_warehouse\_stock(), fetch\_showroom\_stock(), dan fetch\_library\_stock() |
| 8 | get\_book() | Mengambil data semua buku yang ditulis oleh seorang penulis beserta data stok bukunya untuk faktur *showroom*. |
| 9 | get\_book\_dynamic\_stock() | Mengambil data semua buku yang ditulis oleh seorang penulis beserta data stok bukunya untuk semua faktur. |
| 10 | get\_discount() | Mengambil data diskon untuk sebuah jenis *customer*. |
| 11 | get\_book\_royalty() | Mengambil nilai royalti pada sebuah buku. |
| 12 | get\_customer() | Mengambil data seorang *customer* dari database. |
| 13 | get\_last\_invoice\_number() | Mengambil data nomor *invoice* terakhir dan mengembalikan nilai untuk nomor *invoice* yang akan dibuat. |
| 14 | filter\_invoice() | Mengambil seluruh data *invoice* dan total *record* yang ada dari database berdasarkan filter yang dipakai. |
| 15 | when() | Fungsi untuk membantu *query when()* di MySQL pada proses *query* filter\_invoice(). |
| 16 | filter\_book\_request() | Mengambil seluruh data *book\_request* dan total *record* yang ada dari database berdasarkan filter yang dipakai. |
| 17 | when\_request() | Fungsi untuk membantu *query when()* di MySQL pada proses *query* filter\_book\_request(). |
| 18 | start\_progress() | Memperbarui status pada *invoice* menjadi *“preparing”*. |
| 19 | finish\_progress() | Memperbarui status pada *invoice* menjadi *“preparing\_finish”*. |

## Implementasi Fitur Proforma

Proforma merupakan salah satu fitur yang dikembangkan untuk diberikan kepada calon pembeli. Proforma akan dikeluarkan sebelum faktur dikeluarkan, dan setelah disetujui oleh pembeli dan dibayar, maka proforma tersebut akan diubah ke faktur tunai secara otomatis.

Halaman utama proforma juga memiliki struktur yang sama dengan halaman faktur. Untuk filter yang tersedia, pengguna hanya dapat memfilter berdasarkan jenis *customer* dan nama *customer*, karena pada proforma itu sendiri, tidak ada informasi yang berkaitan dengan jenis proforma atau status proforma. Tampilan untuk halaman utama proforma dapat dilihat pada gambar 14.



Gambar Halaman utama proforma

Pada kolom di bagian kanan, pengguna dapat menemui dua tombol, yaitu aksi dan *edit*. Tombol aksi sendiri akan memberikan pengguna dua pilihan, yaitu untuk membuat faktur (mengubah proforma menjadi faktur), atau menghapus proforma. Pada bagian *edit*, pengguna akan dibawa ke halaman untuk mengubah informasi yang berkaitan dengan proforma tersebut. Setelah proforma diubah menjadi faktur, maka data proforma di *database* juga akan dihapus.

Tampilan pada saat pengguna akan menambahkan proforma sendiri juga identik dengan halaman untuk menambahkan faktur, tetapi tanpa pilihan untuk memilih jenis proforma/faktur. Secara *default*, proforma memiliki waktu jatuh tempo H+3 untuk pembayarannya, tetapi pengguna dapat mengubah sesuai yang diinginkan. Pengguna juga dapat mencetak detail dari setiap proforma dalam bentuk PDF dengan mengakses halaman *view* untuk setiap proforma.

Fitur *proforma* diatur melalui file *controller* bernama *Proforma.php* dan *Proforma\_model*.php yang terletak pada modul *proforma*. Tabel **xxx** mendeskripsikan fungsi pada *controller*, sedangkan tabel **xxx** mendeskripsikan fungsi pada *model*.

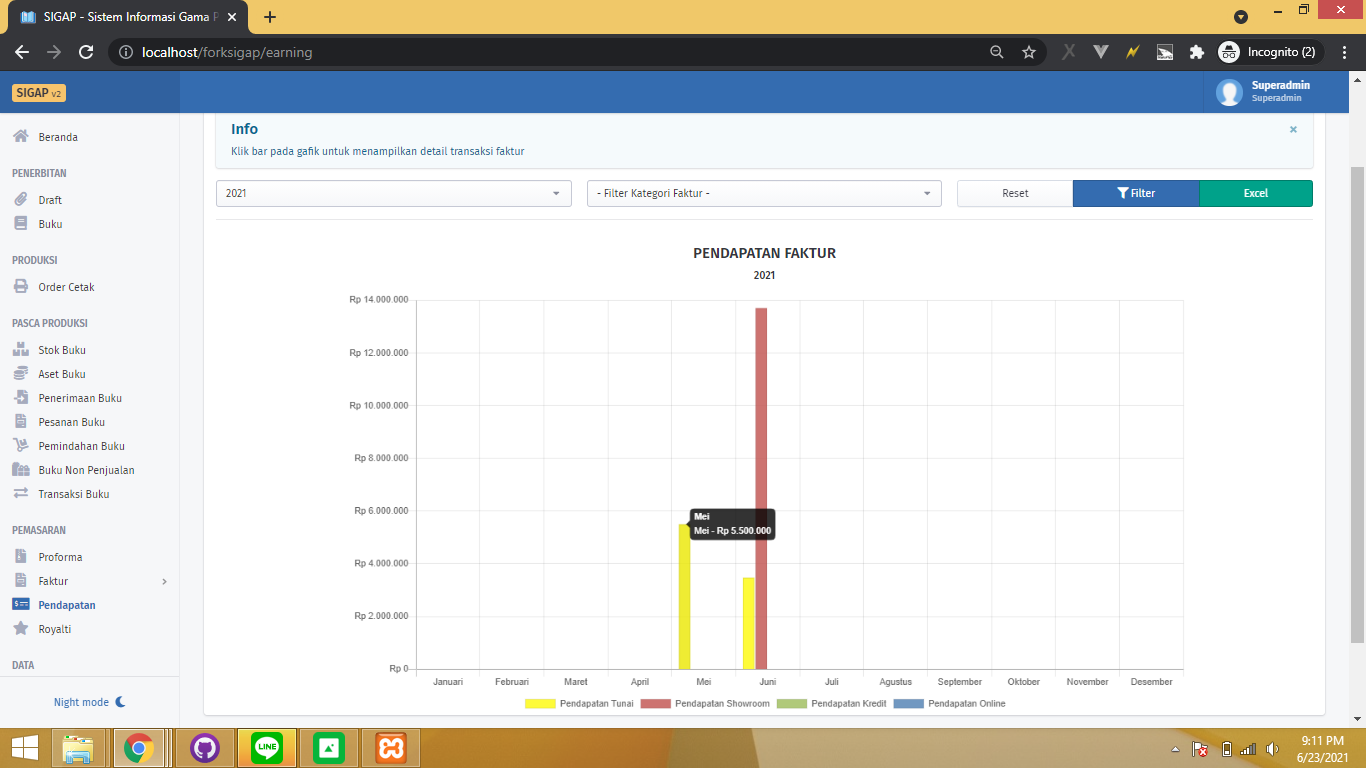
|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | \_\_construct() | Memuat model dan helper yang akan digunakan, yaitu *proforma*\_*model, sales\_helper,* serta model dari modul lain yaitu *book, book\_stock,* dan *book\_transaction*. |
| 2 | index() | Mengarahkan pengguna ke halaman utama *proforma*, dan mengirimkan data *proforma*dari *proforma\_model* ke view. |
| 3 | view() | Mengarahkan pengguna ke halaman sebuah *proforma* dan mengirimkan data *proforma* tersebut ke view. |
| 4 | action() | Memproses *proforma* menjadi *invoice* berjenis tunai, serta menghapus data *proforma* tersebut dari database dan mengarahkan pengguna ke halaman *view invoice* yang baru saja dibuat. |
| 5 | add() | Mengarahkan pengguna ke halaman untuk menambahkan *proforma* baru dan mengirimkan data *post*  untuk menambahkan *proforma*baru ke dalam database. |
| 6 | edit() | Mengarahkan pengguna ke halaman untuk mengubah data *proforma* sebelumnya dan mengirimkan data *post*  untuk memperbarui data *proforma*  ke dalam database. |
| 7 | generate\_pdf() | Membuat file PDF untuk *proforma* yang jenisnya selain *showroom.* |
| 8 | api\_get\_book() | Fungsi API yang digunakan untuk mengirimkan informasi sebuah buku ketika pengguna memilih sebuah buku dari dropdown buku di *view* add *proforma*. |
| 9 | api\_get\_customer() | Fungsi API yang digunakan untuk mengirimkan informasi sebuah *customer* ketika pengguna memilih seorang *customer* dari dropdown *customer* di *view* add *proforma*. |
| 10 | api\_get\_discount() | Fungsi API untuk mengirim informasi diskon dari jenis *customer* yang dipilih pada dropdown *customer* di *view* *proforma*. |

|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | validate\_proforma() | Memvalidasi data *post* ketika akan menambahkan atau memperbarui *proforma* ke database. |
| 2 | fetch\_proforma\_id() | Mengambil data sebuah *proforma* dari database. |
| 3 | fetch\_proforma\_book() | Mengambil data *proforma* dan *proforma\_book* yang berkaitan dengan *proforma\_id* yang sama dari database. |
| 4 | fetch\_book\_info() | Mengambil data judul buku dengan data *book\_id* dari database. |
| 5 | fetch\_warebouse\_stock() | Mengambil data stok sebuah buku yang berada di *warehouse.* |
| 6 | get\_ready\_book\_list() | Mengirimkan data dari fungsi fetch\_showroom\_stock() untuk memastikan stok buku mencukupi untuk mengubah *proforma* menjadi *invoice* tunai*.* |
| 7 | get\_book() | Mengambil data semua buku yang ditulis oleh seorang penulis beserta data stok bukunya. |
| 8 | get\_discount() | Mengambil data diskon untuk sebuah jenis *customer*. |
| 9 | get\_customer() | Mengambil data seorang *customer*. |
| 10 | get\_last\_proforma\_number() | Mengambil data nomor *proforma* terakhir dan mengembalikan nilai untuk nomor *proforma* yang akan dibuat. |
| 11 | filter\_proforma() | Mengambil seluruh data *proforma* dan total *record* yang ada dari database berdasarkan filter yang dipakai. |
| 12 | when() | Fungsi untuk membantu *query when()* di MySQL pada proses *query* filter\_proforma(). |

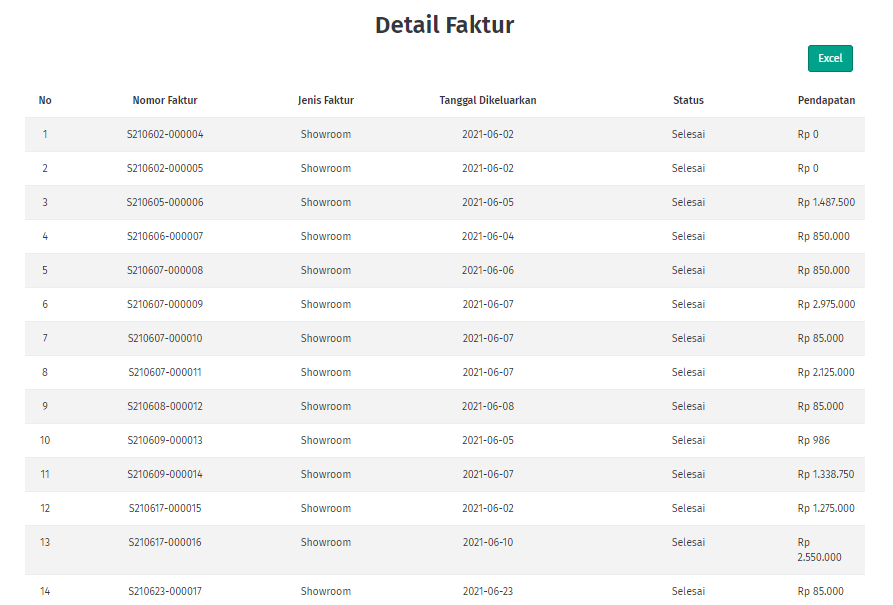
## Implementasi Fitur Pendapatan

Fitur pendapatan berisi halaman yang menampilkan data hasil pendapatan penjualan buku. Fitur ini mengambil data dari tabel faktur di database. Data kemudian dikelompokkan berdasarkan jenis faktur dan dapat di *filter* berdasarkan data per bulan. Data yang telah dikelompokkan kemudian akan dibuat dalam bentuk diagram batang dan diagram lingkaran menggunakan *library* chart js.

Pada halaman utama fitur pendapatan, pengguna dapat melihat grafik hasil rekap data faktur dalam bentuk diagram batang. Saat pengguna menekan diagram, maka akan muncul tabel detail pendapatan yang berisi nomor faktur dan total pendapatan dari setiap faktur seperti pada Gambar 15 dan Gambar 16

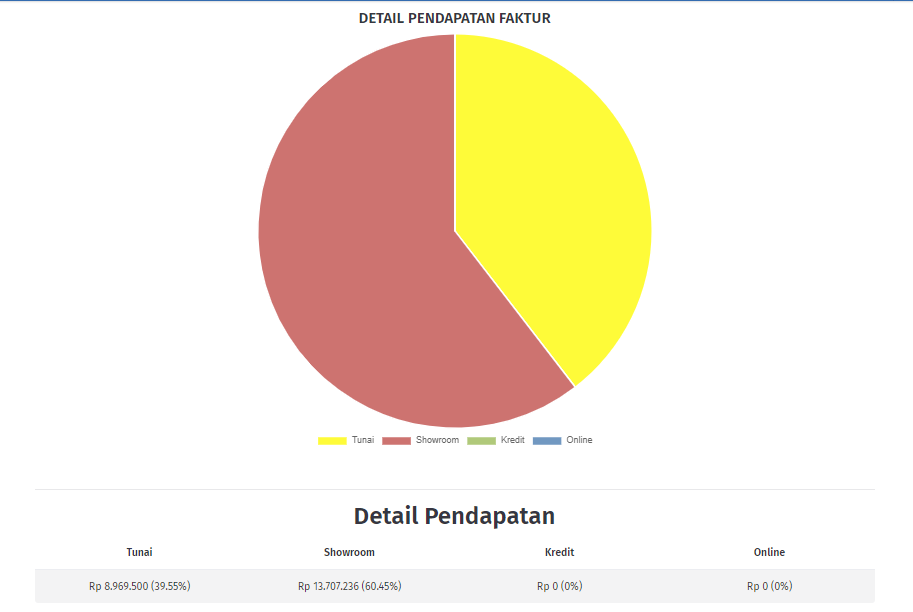


Gambar Diagram batang data pendapatan faktur



Gambar 16 Tampilan tabel detail pendapatan faktur

Data pendapatan juga dapat ditampilkan dalam bentuk diagram lingkaran. Gambar 17 menunjukkan tampilan data pendapatan dalam bentuk diagram lingkaran.

****

Gambar Diagram lingkaran data pendapatan faktur

Fitur *earning* diatur melalui file *controller* bernama *Earning.php* dan *Earning\_model*.php yang terletak pada modul *earning*. Tabel **xxx** mendeskripsikan fungsi pada *controller*, sedangkan tabel **xxx** mendeskripsikan fungsi pada *model*.

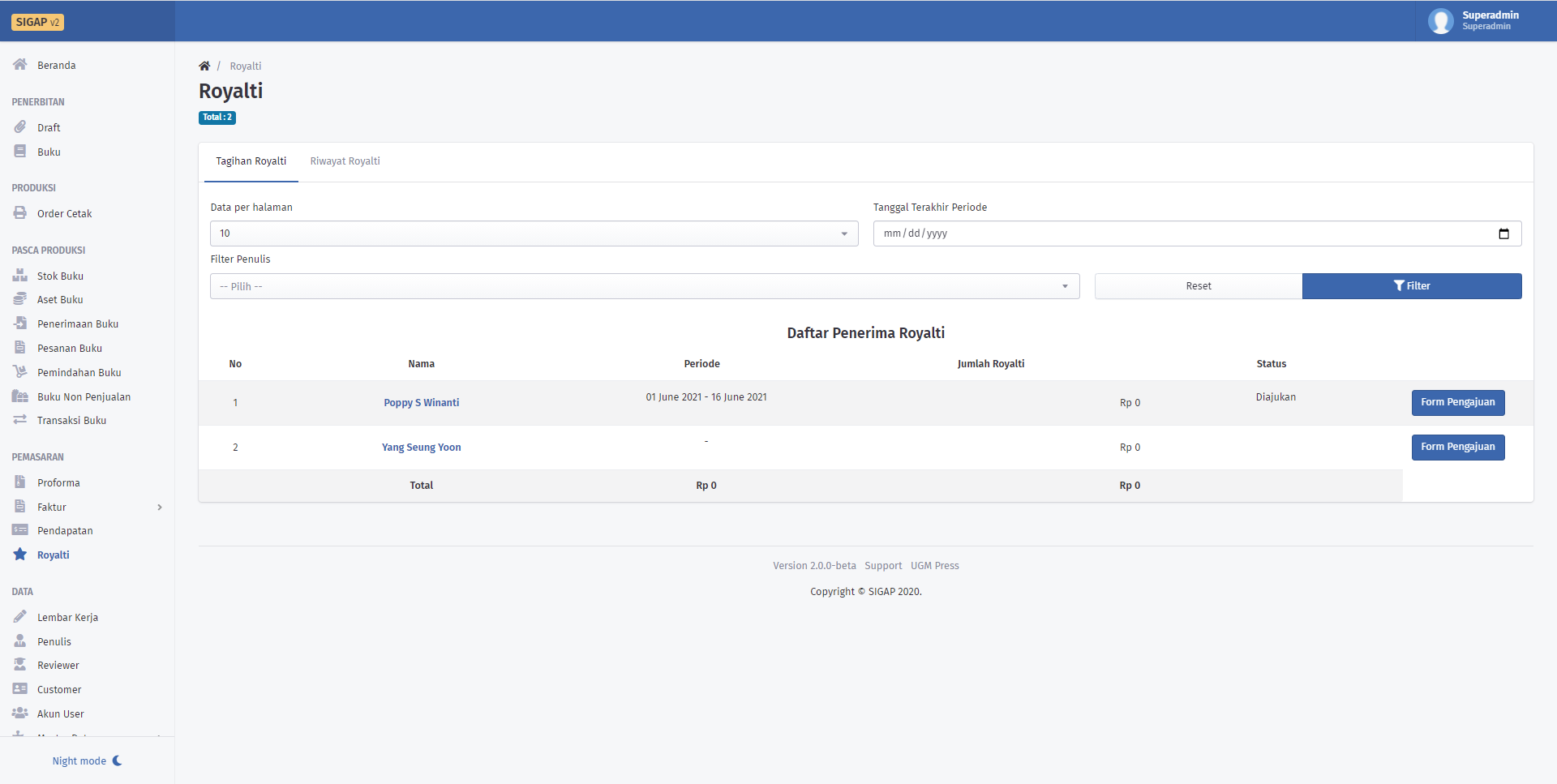
|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | \_\_construct() | Memuat model dan helper yang akan digunakan, yaitu *earning*\_*model* dan *sales\_helper.* |
| 2 | index() | Mengarahkan pengguna ke halaman *“Pendapatan Faktur”*, dan mengirimkan data *earning* dari *earning\_model* ke view. Data pada halaman ini akan digunakan untuk pembuatan grafik batang pendapatan dalam satu tahun yang dibagi berdasarkan bulan dan jenis fakturnya. |
| 3 | detail() | Mengarahkan pengguna ke halaman *“Detail Pendapatan”*, dan mengirimkan data *earning* dari *earning\_model* ke view. Data pada halaman ini akan digunakan untuk pembuatan grafik *pie* pendapatan dalam satu tahun/bulan dan menunjukkan pendapatan untuk setiap jenis faktur. |
| 4 | api\_get\_invoice() | API untuk mengirimkan data yang berkaitan dengan pendapatan pada sebuah jenis faktur pada periode tertentu (berdasarkan filter) untuk pembuatan tabel di *view*. |
| 5 | call\_generate\_excel() | API untuk memanggil fungsi generate\_excel. Fungsi ini dipanggil ketika pengguna menekan tombol pada tabel yang dibuat dari data api\_get\_invoice() |
| 6 | generate\_excel() | Membuat file excel berdasarkan filter dan jenis menunya. |

|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | filter\_total() | Mengambil data pendapatan berdasarkan filter yang dipakai. Data pendapatan diambil dari penjualan dari seluruh *invoice* pada periode filter tersebut dan dibagi berdasarkan jenis fakturnya. |
| 2 | filter\_detail() | Mengambil data total pendapatan berdasarkan filter yang dipakai. Data total hanya mengambil total penjualan berdasarkan periode dan jenis fakturnya. |
| 3 | get\_invoice() | Mengambil data *invoice* untuk dikembalikan ke fungsi api\_get\_invoice() di *controller*. |
| 4 | filter\_excel\_total() | Mengambil data pendapatan untuk membuat file excel dari *controller* index(). |
| 5 | filter\_excel\_detail() | Mengambil data pendapatan untuk membuat file excel dari *controller* detail() dan call\_generate\_excel(). |

## Implementasi Fitur Royalti

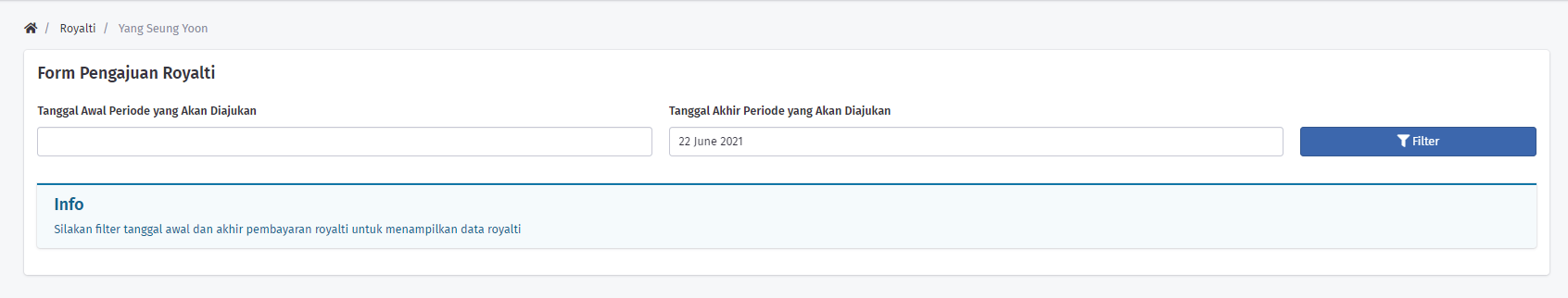
Pada halaman utama fitur royalti, pengguna dapat melihat bahwa terdapat dua *tab,* masing-masinguntuk mengakses halaman tagihan royalti dan riwayat royalti. Pada halaman tagihan royalti, pengguna dapat melihat informasi mengenai royalti yang belum dibayar atau masih dalam tahap diajukan. Pengguna dapat memfilter berdasarkan tanggal terakhir periode dan nama penulis saja. Apabila tidak ada filter yang digunakan, maka halaman tagihan royalti akan menampilkan seluruh royalti hingga H-1 yang masih dalam tahap diajukan atau belum dibayar. Filter berdasarkan tanggal terakhir periode juga dibatasi hingga H-1, untuk mencegah potensi adanya royalti pada hari tersebut dan sehingga tidak termasuk ke dalam tagihan royalti berikutnya.

Data yang ditampilkan pada halaman royalti memanfaatkan data dari tabel royalti dan juga proses *join* dari tabel-tabel lainnya, yaitu tabel faktur dan tabel *draft author*. Tetapi, pada halaman tagihan royalti itu sendiri, hanya informasi mengenai nama penulis, periode royalti yang belum dibayar/masih diajukan, jumlah royalti, serta statusnya. Untuk melihat detail dari royalti, pengguna dapat menekan tombol form pengajuan. Gambar 18 merupakan tampilan halaman utama fitur royalti.



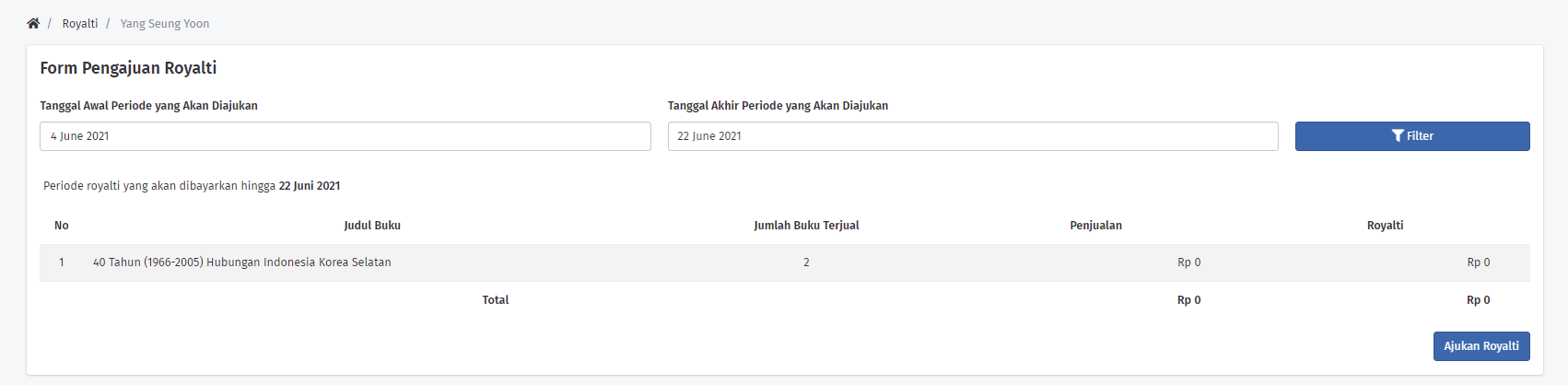
Gambar Halaman utama fitur “Royalti”

Pada halaman *form* pengajuan, terdapat beberapa tampilan berbeda yang akan tampil pada halaman pengguna. Apabila tagihan penulis bukan dalam status diajukan, maka pengguna akan dapat melihat sesuai dengan tampilan pada Gambar 19. Pengguna diharuskan memilih tanggal mulai hingga akhir periode yang ingin dibayar royaltinya. Apabila penulis tersebut sudah pernah menerima royalti, maka tanggal awal periode yang akan diajukan akan secara otomatis terisi, dengan nilai yaitu tanggal akhir periode pada tagihan royalti sebelumnya ditambah dengan satu hari.



Gambar 19 Tampilan form pengajuan royalti saat pengguna mengajukan pertama kali

Setelah di-*filter*, maka pengguna akan melihat informasi mengenai buku yang terjual, pendapatan kotor yang diperoleh, serta royalti yang perlu dibayarkan pada penulis. Gambar 20 menunjukkan tampilan halaman pengajuan royalti setelah di-*filter*.



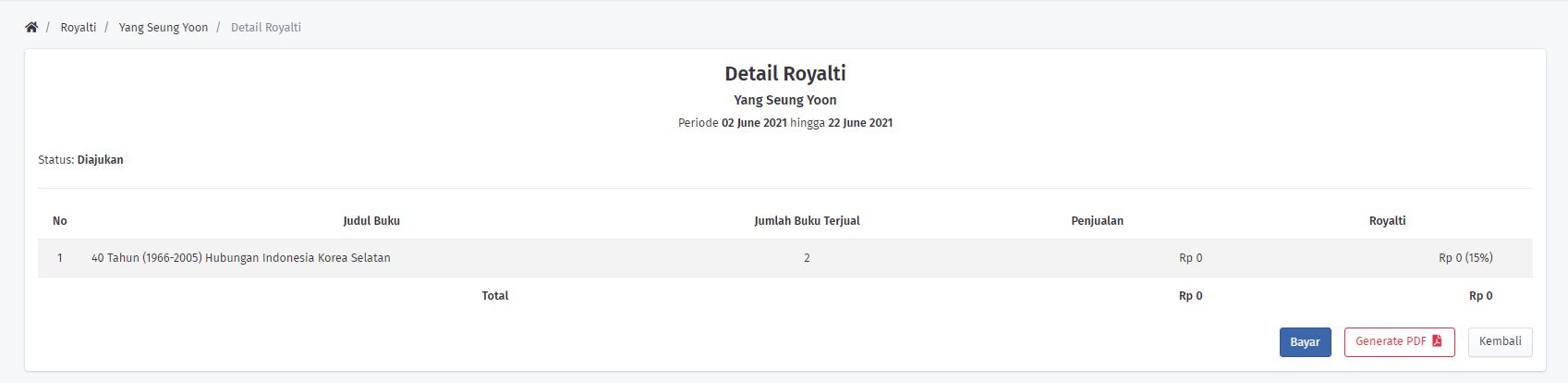
Gambar 20 Tampilan halaman pengajuan royalti setelah di-*filter*

Jika royalti sedang diajukan, maka pada halaman pengajuan royalti akan ditampilkan data tagihan royalti. Pada Gambar 21, pengguna dapat melihat detail dari tagihan royalti tersebut, atau langsung mengkonfirmasi pembayaran dengan menekan tombol “bayar”**.**

****

Gambar 21 Tampilan halaman pengajuan royalti setelah mengajukan royalti

Tombol detail seperti yang sebelumnya dijelaskan, akan menunjukkan detail dari tagihan royalti yang perlu dibayar atau sudah pernah dibayar. Selain melalui halaman form pengajuan, halaman detail royalti ini dapat diakses melalui tombol di halaman riwayat royalti. Pada halaman ini, pengguna dapat melihat informasi detail royalti. Pengguna juga dapat mengonfirmasi pembayaran tagihan royalti atau mencetak PDF tagihan royalti. Gambar 22 menunjukkan tampilan halaman detail royalti.



Gambar Halaman detail royalti

Halaman terakhir pada fitur royalti adalah riwayat royalti. Halaman ini akan mencatat seluruh royalti yang telah selesai dibayar dan sedang diajukan. Struktur tabel pada halaman ini juga menyerupai informasi pada tabel di halaman tagihan royalti, tetapi dengan kolom tambahan berupa status royalti (diajukan atau dibayar), serta tanggal dibayar. Terdapat tambahan *filter* untuk mencari royalti yang dibuat pada tanggal tertentu, pengguna mem-*filter* dengan memilih tanggal awal dan tanggal akhir royalti dibuat. Informasi pada tabel royalti diurutkan berdasarkan tagihan royalti terbaru.

Fitur royalti diatur melalui file *controller* bernama *Royalty.php* dan *Royalty\_model*.php yang terletak pada modul *royalty*. Tabel **xxx** mendeskripsikan fungsi pada *controller*, sedangkan tabel **xxx** mendeskripsikan fungsi pada *model*.

|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | \_\_construct() | Memuat model dan helper yang akan digunakan, yaitu *royalty*\_*model, book\_model,* dan *sales\_helper.* |
| 2 | index() | Mengarahkan pengguna ke halaman utamaroyalti, dan mengirimkan data royalti dari *Royalty\_model* ke view. Data yang dikirimkan adalah royalti yang masih diajukan atau belum dibayar. |
| 3 | history() | Mengarahkan pengguna ke halaman riwayatroyalti, dan mengirimkan data royalti dari *Royalty\_model* ke view. Data yang dikirimkan adalah royalti yang dalam tahap diajukan dan yang sudah selesai dibayar. |
| 4 | view() | Mengarahkan pengguna ke form pengajuan royalti. Apabila sudah ada royalti yang sedang diajukan, maka yang ditampilkan adalah pilihan untuk menyelesaikan pembayaran royalti tersebut. |
| 5 | view\_detail() | Mengarahkan pengguna ke halaman detail sebuah royalti dan data yang berkaitan dengan royalti tersebut. |
| 6 | generate\_pdf() | Membuat file PDF untuk sebuah royalti. |
| 7 | pay() | Memperbarui status pembayaran royalti. Apabila belum ada royalti sebelumnya atau status royalti terakhir adalah dibayar, maka membuat *record* royalti baru. Apabila sudah ada dan dalam tahap diajukan, maka memperbarui status royalti yang ada menjadi selesai dibayar. |
| 8 | filter\_author() | Fungsi untuk menghapus data penulis yang tidak memiliki tagihan royalti pada periode tersebut dari array di index(). |

|  |  |  |
| --- | --- | --- |
| No. | Nama Fungsi | Keterangan |
| 1 | validate\_royalty() | Memvalidasi input ketika pengguna ingin mengajukan royalti seorang penulis untuk pertama kalinya. |
| 2 | get\_dropdown\_author\_history() | Mengambil data royalti penulis untuk sebagai list pada dropdown pada halaman riwayat royalti. |
| 3 | fetch\_royalty\_history() | Mengambil data seluruh royalti yang pernah diajukan atau dibayar untuk seorang penulis. |
| 4 | fetch\_latest\_royalty() | Mengambil data royalti terakhir yang pernah diajukan atau dibayar untuk seorang penulis. |
| 5 | fetch\_all\_royalty\_history() | Mengambil data seluruh royalti yang pernah diajukan atau dibayar berdasarkan filter yang dipilih. |
| 6 | author\_earning() | Mengambil data seluruh royalti penulis beserta informasi *invoice* yang berkaitan berdasarkan filter yang dipilih. |
| 7 | author\_details() | Mengambil data royalti seorang penulis beserta informasi *invoice* dan buku yang ditulis. |
| 8 | stocks\_info() | Mengambil data informasi stok buku sekarang, serta stok buku yang hilang berdasarkan penjualan *invoice* pada periode tertentu untuk perhitungan pada PDF royalti. |
| 9 | get\_non\_sales\_book() | Mengambil data jumlah stok buku yang berkurang karena non penjualan pada periode ternteu untuk perhitungan pada PDF royalti. |

**BAB II**

# PERUBAHAN PADA PROSES DESAIN DAN IMPLEMENTASI

## *Requirements* Proyek Capstone

Proses implementasi proyek capstone ini berjalan sesuai dengan perencanaan dan desain pada dokumen C250 dengan beberapa perubahan dan tambahan. Berikut merupakan rincian mengenai fitur yang telah memenuhi kebutuhan yang didefinisikan dan fitur tambahan sesuai dengan tambahan dari klien.

### Requirements Awal

*Requirements* yang telah didefinisikan pada saat tahapan perancangan proyek capstone ini (Saat C250) adalah sebagai berikut.

1. Dapat membuat seluruh jenis faktur yang ada di UGM Press
2. Melihat rekapan data faktur yang sudah dibuat sebelumnya pada satu halaman
3. Mengubah data faktur pada *website* ke dalam format PDF sesuai format faktur yang sudah ada dan dapat diunduh guna keperluan laporan pemasaran atau lainnya
4. Melihat seluruh royalti penulis yang belum dibayar oleh UGM Press pada satu halaman
5. Dapat melakukan perubahan status penerimaan royalti penulis (Status “Diajukan” dan status “Sudah Dibayar”) guna keperluan rekap data
6. Melihat seluruh riwayat royalti semua penulis pada satu halaman
7. Dapat melihat riwayat royalti suatu penulis tertentu
8. Besar royalti ditentukan secara otomatis oleh sistem berdasarkan jumlah seluruh penjualan buku yang ditulis oleh penulis dari keseluruhan jenis faktur yang dibuat
9. Dapat mengunduh laporan royalti suatu penulis dalam format PDF sesuai format laporan royalti yang sudah ada

### *Requirements* Tambahan

Pada saat proses pengembangan sistem, terdapat beberapa tambahan fitur guna mempermudah pekerjaan dan menambah efektivitas sistem. Tambahan fitur tersebut adalah sebagai berikut.

1. Proforma

Proforma merupakan *requirement* tambahan yang berfungsi sebagai simulasi dan penawaran kepada calon *customer* UGM Press. Buku yang dimasukkan ke dalam proforma tidak mengurangi stok yang ada tetapi pembuatan proforma harus mengacu kepada jumlah buku sesuai stok yang ada saat itu. *Requirements* untuk fitur proforma ini adalah sebagai berikut.

* Dapat membuat proforma baru dengan buku yang sesuai dengan ketersediaan stok saat proforma dibuat
* Mengunduh proforma dalam format PDF agar dapat dikirim kepada calon *customer*
* Dapat mengubah proforma menjadi faktur dengan jenis tunai ketika *customer* menyetujui penawaran dan melakukan pembayaran sesuai dengan pesanan yang ada di proforma

1. Pendapatan

Fitur ini berguna untuk menampilkan pendapatan faktur pada periode tertentu. Fitur ini diminta karena penting untuk keperluan bisnis UGM Press untuk kedepannya, karena dengan fitur ini diharapkan dapat dijadikan acuan atau pertimbangan terkait pembuatan kebijakan UGM Press terkait pemasaran. *Requirements* terkait fitur ini adalah sebagai berikut.

* Dapat melihat jumlah pendapatan masing-masing jenis faktur dalam bentuk grafik
* Dapat melihat jumlah pendapatan masing-masing faktur pada satu jenis faktur tertentu dalam bentuk tabel
* Mengunduh tabel pendapatan dalam bentuk Excel
* Melihat perbandingan jumlah pendapatan seluruh jenis faktur dalam bentuk diagram lingkaran

1. *Customer*

Fitur *customer* dibutuhkan guna mengisi data pelanggan dengan lebih cepat pada saat pembuatan faktur atau proforma. Dengan menyimpan data *customer*, memungkinkan untuk menyortir proforma atau faktur berdasarkan nama pelanggannya. Fitur ini diajukan dengan beberapa *requirements* sebagai berikut.

* Menambah dan mengedit data *customer*
* Melihat seluruh data *customer* pada satu halaman
* Mengubah besarnya diskon yang didapatkan sesuai dengan jenis *customer*

**BAB III**

# PENGUJIAN DAN ANALISIS

## Metode Pengujian

### Back-end

Metode pengujian yang dilakukan untuk bagian *back*-*end* adalah metode *whitebox testing*. Metode ini melakukan pengujian suatu sistem atau *software* yang dilakukan oleh *software developer* atau pengembang untuk memeriksa dan menganalisa kode program yang dibuat terdapat kesalahan atau tidak dan dapat berfungsi sesuai dengan yang diharapkan. Pengujian ini membutuhkan penguji yang mengetahui secara penuh mengenai cara kerja *source code* program khususnya pada logika dan struktur kode. Pengujian ini berfokus pada *source code* program dan mengabaikan tampilan atau *user interface* sistem.

### Front-end

Metode pengujian yang dilakukan untuk proyek capstone ini khususnya pada bagian *front*-*end* adalah metode *blackbox testing.* *Blackbox testing* adalah metode pengujian yang berfokus pada aspek fungsionalitas sistem dan kesesuaian dengan alur bisnis yang diinginkan oleh calon pengguna sistem. Pengujian ini tidak melihat dan menguji *source code* program dan berfokus pada tampilan luar (*interface*). Metode pengujian ini dilakukan dengan memberikan penguji serangkaian tugas dengan input dan alur yang sudah ditentukan. Penguji kemudian diminta untuk melaksanakan tugas dan melaporkan apakah hasil yang didapatkan sesuai dengan hasil yang diharapkan.

### User Experience

Metode pengujian untuk *user experience* menggunakan evaluasi SUS (*System Usability Scale*) untuk melakukan *usability testing*. SUS merupakan salah satu alat pengujian *usability* yang dikembangkann oleh John Brooke pada tahun 1986. SUS dapat digunakan untuk menilai kegunaan berbagai produk dan layanan. Penulis menggunakan SUS karena responden dapat dengan cepat dan mudah menjawab pertanyaan dan SUS dapat menjawab pertanyaan terkait kelayakan sistem pemasaran SIGAP yang dikembangkan.

**3.1.4 Pengujian Sistem Pemasaran SIGAP**

Sistem pemasaran SIGAP merupakan pengembangan dari sistem informasi SIGAP sebelumnya dan berfungsi untuk mengotomasi pekerjaan bagian pemasaran UGM Press. Sistem SIGAP yang terbaru diujikan kepada karyawan bagian pemasaran UGM Press sebagai pengguna utama sistem SIGAP yang penulis kembangkan. Pengujian ini dimaksudkan untuk melihat perbandingan kinerja bagian pemasaran UGM Press dengan mengerjakan pekerjaannya saat ini dibandingkan dengan menggunakan sistem SIGAP. Pengujian dilakukan oleh karyawan UGM Press bagian pemasaran secara langsung di UGM Press, Jl. Sendok, Karanggayam, Sleman, Yogyakarta. Penulis memberikan beberapa skenario untuk dilakukan pengujian kepada pengguna kemudian setelah skenario selesai dijalankan, pengguna diminta untuk menulis pendapat atau *feedback* terhadap sistem pemasaran SIGAP.

**3.2. Proses Pengujian**

**3.2.1. *Back-end***

**3.2.2. *Front-end***

Pengujian untuk bagian *front*-*end* dilakukan dengan menguji setiap fungsi yang ada pada tampilan (*interface*). Penguji diminta untuk melakukan setiap skenario yang mungkin terjadi dalam alur bisnis dan menguji apakah keluaran yang dihasilkan sesuai dengan keluaran yang diharapkan. Berikut merupakan *test case* yang dilakukan :

Tabel *Test Case* Halaman *Index* Proforma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan keseluruhan data proforma yang ada di *database* |  | Menampilkan nomor urut, nomor proforma, nama *customer*, jenis *customer*, tanggal dibuatnya proforma, dan jatuh tempo proforma | Menampilkan nomor, nomor proforma, nama *customer*, jenis *customer*, tanggal dibuatnya proforma, dan jatuh tempo proforma | Valid |
| Mengubah jumlah data yang ditampilkan pada halaman *index* | 10  25  50  100 | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Valid |
| Menyortir dan menampilkan jenis *customer* tertentu | Distributor  Reseller  Penulis  Member  Umum | Menunjukkan data yang dipilih sesuai dengan masukkan | Menunjukkan data yang dipilih sesuai dengan masukkan | Valid |
| Melakukan pencarian data sesuai dengan masukkan pengguna | Data berbentuk teks | Menunjukkan data sesuai dengan masukkan | Menunjukkan data sesuai dengan masukkan | Valid |
| Menampilkan dan me-*reset* data yang ditampilkan seperti awal | Menekan tombol “Reset” | Menampilkan semua data yang ada di *database* proforma | Menampilkan semua data yang ada di *database* proforma | Valid |
| Berpindah ke halaman detail proforma | Menekan nomor proforma pada tabel | Menuju ke halaman detail proforma sesuai dengan nomor yang ditekan | Menuju ke halaman detail proforma sesuai dengan nomor yang ditekan | Valid |
| Berpindah ke halaman tambah proforma | Menekan tombol “Tambah” | Menuju ke halaman tambah proforma | Menuju ke halaman tambah proforma | Valid |
| Mengubah proforma menjadi faktur tunai | Menekan tombol “Aksi” kemudian pilih opsi “Buat Faktur” | Menuju ke halaman detail faktur | Menuju ke halaman detail faktur | Valid |
| Menghapus proforma | Menekan tombol “Aksi” kemudian pilih opsi “Hapus Proforma” | Muncul peringatan sukses berbentuk *toast* dan proforma terhapus dari halaman *index* | Muncul peringatan sukses berbentuk *toast* dan proforma terhapus dari halaman *index* | Valid |
| Mengubah detail proforma | Menekan tombol dengan *icon* pensil | Menuju ke halaman edit proforma | Menuju ke halaman edit proforma | Valid |

Tabel *Test Case* Halaman TambahProforma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menambah proforma dengan memilih *customer* dari *database* | Tanggal jatuh tempo  *Customer*  Buku | Muncul peringatan sukses berbentuk *toast* dan berpindah ke halaman *index* proforma | Muncul peringatan sukses berbentuk *toast* dan berpindah ke halaman *index* proforma | Valid |
| Menambah proforma dengan *customer* baru/belum ada di *database* | Tanggal jatuh tempo  Nama *Customer*  Alamat *Customer*  Nomor Telepon *Customer*  Jenis *Customer*  Buku | Muncul peringatan sukses berbentuk *toast* dan berpindah ke halaman *index* proforma | Muncul peringatan sukses berbentuk *toast* dan berpindah ke halaman *index* proforma | Valid |
| Menambah proforma dengan data yang kurang | Menekan tombol Submit dengan data yang kurang/belum diisi secara keseluruhan | Muncul peringatan di bawah kolom data yang kosong | Muncul peringatan di bawah kolom data yang kosong | Valid |
| Menambah proforma dengan jumlah buku yang melebihi stok | Memilih buku dan menekan tombol “Tambah Barang” dengan jumlah buku yang melebihi stok | Muncul peringatan dan buku tidak masuk ke dalam pesanan | Muncul peringatan dan buku tidak masuk ke dalam pesanan | Valid |
| Menghapus buku dalam tabel pesanan | Menekan tombol “Hapus” sesuai buku yang ingin dihilangkan dalam tabel pesanan | Buku terhapus dalam tabel pesanan | Buku terhapus dalam tabel pesanan | Valid |
| Kembali ke halaman sebelumnya (Halaman *Index* Proforma) | Menekan tombol “Back” atau teks “Proforma” pada bagian atas halaman | Menuju ke halaman *Index* Proforma | Menuju ke halaman *Index* Proforma | Valid |

Tabel *Test Case* Halaman DetailProforma

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Melihat detail suatu proforma |  | Menampilkan nomor proforma, nama *customer*, nomor telepon *customer*, tanggal jatuh tempo, tanggal dibuat proforma, dan semua buku yang dipilih | Menampilkan nomor proforma, nama *customer*, nomor telepon *customer*, tanggal jatuh tempo, tanggal dibuat proforma, dan semua buku yang dipilih | Valid |
| Mengubah proforma menjadi faktur tunai | Menekan tombol “Buat Faktur” kemudian menekan tombol “Confirm” pada modal | Muncul modal untuk menyetujui kemudian menuju ke halaman detail faktur | Muncul modal untuk menyetujui kemudian menuju ke halaman detail faktur | Valid |
| Mengunduh proforma | Menekan tombol “Generate PDF” | PDF Proforma terunduh | PDF Proforma terunduh | Valid |

Tabel *Test Case* Halaman *Dashboard* Faktur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan keseluruhan data faktur yang ada di *database* |  | Menampilkan nomor urut, nomor faktur, jenis faktur, nama *customer*, jenis *customer*, tanggal dibuatnya faktur, jatuh tempo faktur, dan status faktur | Menampilkan nomor urut, nomor faktur, jenis faktur, nama *customer*, jenis *customer*, tanggal dibuatnya faktur, jatuh tempo faktur, dan status faktur | Valid |
| Mengubah jumlah data yang ditampilkan pada halaman *index* | 10  25  50  100 | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Valid |
| Menyortir dan menampilkan jenis fakturtertentu | Kredit  Tunai  *Online*  *Showroom* | Menunjukkan data yang dipilih sesuai dengan masukkan | Menunjukkan data yang dipilih sesuai dengan masukkan | Valid |
| Menyortir dan menampilkan jenis *customer* tertentu | Distributor  Reseller  Penulis  Member  Umum | Menunjukkan data yang dipilih sesuai dengan masukkan | Menunjukkan data yang dipilih sesuai dengan masukkan | Valid |
| Menyortir dan menampilkan jenis *customer* tertentu | Belum Konfirmasi  Sudah Konfirmasi  Diproses  Siap Diambil  Selesai  DIbatalkan | Menunjukkan data yang dipilih sesuai dengan masukkan | Menunjukkan data yang dipilih sesuai dengan masukkan | Valid |
| Melakukan pencarian data sesuai dengan masukkan pengguna | Data berbentuk teks | Menunjukkan data sesuai dengan masukkan | Menunjukkan data sesuai dengan masukkan | Valid |
| Menampilkan dan me-*reset* data yang ditampilkan seperti awal | Menekan tombol “Reset” | Menampilkan semua data yang ada di *database* proforma | Menampilkan semua data yang ada di *database* proforma | Valid |
| Berpindah ke halaman detail faktur | Menekan nomor faktur pada tabel | Menuju ke halaman detail faktur sesuai dengan nomor yang ditekan | Menuju ke halaman detail faktur sesuai dengan nomor yang ditekan | Valid |
| Berpindah ke halaman tambah faktur | Menekan tombol “Tambah” | Menuju ke halaman tambah faktur | Menuju ke halaman tambah faktur | Valid |
| Menyetujui faktur yang diajukan ketika pelanggan sudah konfirmasi pembelian | Menekan tombol “Aksi” kemudian pilih opsi “Setuju” | Muncul peringatan sukses berbentuk *toast* dan status faktur berubah menjadi “Sudah Konfirmasi” | Muncul peringatan sukses berbentuk *toast* dan status faktur berubah menjadi “Sudah Konfirmasi” | Valid |
| Menolak faktur yang diajukan | Menekan tombol “Aksi” kemudian pilih opsi “Tolak” | Muncul peringatan sukses berbentuk *toast* dan status faktur berubah menjadi “Dibatalkan” | Muncul peringatan sukses berbentuk *toast* dan status faktur berubah menjadi “Dibatalkan” | Valid |
| Konfirmasi pesanan sudah sudah selesai sesuai faktur | Menekan tombol “Selesai” kemudian menekan tombol “Selesai” pada modal | Muncul peringatan sukses berbentuk *toast* dan status faktur berubah menjadi “Selesai” | Muncul peringatan sukses berbentuk *toast* dan status faktur berubah menjadi “Selesai” | Valid |
| Mengubah detail faktur | Menekan tombol dengan *icon* pensil | Menuju ke halaman edit faktur | Menuju ke halaman edit faktur | Valid |
| Mengunduh tabel faktur | Menekan tombol “Excel” | Tabel faktur dalam bentuk Excel terunduh | Tabel faktur dalam bentuk Excel terunduh | Valid |

Tabel *Test Case* Halaman Tambah Faktur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menambah faktur dengan memilih *customer* dari *database* | Jenis Faktur  Tanggal jatuh tempo  *Customer*  Buku | Muncul peringatan sukses berbentuk *toast* dan berpindah ke halaman *index* faktur | Muncul peringatan sukses berbentuk *toast* dan berpindah ke halaman *index* faktur | Valid |
| Menambah faktur dengan *customer* baru/belum ada di *database* | Jenis Faktur  Tanggal jatuh tempo  Nama *Customer*  Alamat *Customer*  Nomor Telepon *Customer*  Email *Customer*  Jenis *Customer*  Buku | Muncul peringatan sukses berbentuk *toast* dan berpindah ke halaman *index* faktur | Muncul peringatan sukses berbentuk *toast* dan berpindah ke halaman *index* faktur | Valid |
| Menambah faktur dengan jumlah buku yang melebihi stok | Memilih buku dan menekan tombol “Tambah Barang” dengan jumlah buku yang melebihi stok | Muncul peringatan dan buku tidak masuk ke dalam pesanan | Muncul peringatan dan buku tidak masuk ke dalam pesanan | Valid |
| Menambah faktur dengan data yang kurang | Menekan tombol Submit dengan data yang kurang/belum diisi secara keseluruhan | Muncul peringatan di bawah kolom data yang kosong | Muncul peringatan di bawah kolom data yang kosong | Valid |
| Menghapus buku dalam tabel pesanan | Menekan tombol “Hapus” sesuai buku yang ingin dihilangkan dalam tabel pesanan | Buku terhapus dalam tabel pesanan | Buku terhapus dalam tabel pesanan | Valid |
| Kembali ke halaman sebelumnya (Halaman *Index* Faktur) | Menekan tombol “Back” atau teks “Faktur” pada bagian atas halaman | Menuju ke halaman *Dashboard* Faktur | Menuju ke halaman *Dashboard* Faktur | Valid |

Tabel *Test Case* Halaman Edit Faktur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan data suatu faktur yang akan diedit |  | Menampilkan nomor faktur, jenis faktur, asal faktur, nama perpustakaan, tanggal jatuh tempo, identitas *customer*, dan semua buku yang dipilih | Menampilkan nomor faktur, jenis faktur, asal faktur, nama perpustakaan, tanggal jatuh tempo, identitas *customer*, dan semua buku yang dipilih | Valid |
| Mengubah tanggal jatuh tempo faktur | Memilih tanggal pada kolom “Jatuh Tempo” kemudian menekan tombol “Edit” | Tanggal jatuh tempo berubah sesuai dengan yang ditentukan | Tanggal jatuh tempo berubah sesuai dengan yang ditentukan | Valid |
| Mengubah identitas *customer* | Memilih *customer* dari *database* atau menambah identitas *customer* baru kemudian mengisi identitas pada kolom yang disediakan kemudian menekan tombol “Edit” | Identitas *customer* pada faktur berubah | Identitas *customer* pada faktur berubah | Valid |
| Menambah buku dalam pesanan | Memilih judul buku dan jumlah yang diinginkan kemudian menekan tombol “Edit” | Pesanan buku akan bertambah pada faktur | Pesanan buku akan bertambah pada faktur | Valid |
| Menghapus buku dalam pesanan | Menekan tombol “Hapus” sesuai buku yang ingin dihilangkan dalam tabel pesanan kemudian menekan tombol “Edit” | Buku yang dipilih terhapus pada faktur | Buku yang dipilih terhapus pada faktur | Valid |
| Kembali ke halaman sebelumnya (Halaman *Index* Faktur) | Menekan tombol “Back” atau teks “Faktur” pada bagian atas halaman | Menuju ke halaman *Dashboard* Faktur | Menuju ke halaman *Dashboard* Faktur | Valid |

Tabel *Test Case* Halaman Detail Faktur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Melihat detail suatu faktur dengan jenis Tunai, Kredit, dan *Online* |  | Menampilkan nomor faktur, jenis faktur, nama *customer,* nomor *customer*, tanggal jatuh tempo faktur, total berat buku, total ongkos kirim, bukti bayar, tanggal dibuatnya faktur, status faktur, dan semua buku yang dipesan | Menampilkan nomor faktur, jenis faktur, nama *customer,* nomor *customer*, tanggal jatuh tempo faktur, total berat buku, total ongkos kirim, bukti bayar, tanggal dibuatnya faktur, status faktur, dan semua buku yang dipesan | Valid |
| Melihat detail suatu faktur dengan jenis *Showroom* |  | Menampilkan nomor faktur, jenis faktur, nama *customer*, nomor *customer*, tanggal dibuatnya faktur, dan semua buku yang dibeli | Menampilkan nomor faktur, jenis faktur, nama *customer*, nomor *customer*, tanggal dibuatnya faktur, dan semua buku yang dibeli | Valid |
| Melihat detail status suatu faktur jenis Tunai, Kredit, dan *Online* | Menekan tombol “Detail” | Menampilkan tanggal konfirmasi faktur, tanggal faktur mulai diproses, tanggal faktur selesai diproses, dan tanggal faktur diambil oleh pemasaran | Menampilkan tanggal konfirmasi faktur, tanggal faktur mulai diproses, tanggal faktur selesai diproses, dan tanggal faktur diambil oleh pemasaran | Valid |
| Mengunduh faktur dengan jenis Tunai dan Kredit ketika total ongkos kirim atau bukti bayar kosong | Menekan tombol “Generate PDF” kemudian muncul modal untuk mengisikan data total ongkos kirim dan bukti bayar kemudian menekan tombol “Save” | PDF Faktur terunduh | PDF Faktur terunduh | Valid |
| Mengunduh faktur dengan jenis Tunai dan Kredit ketika total ongkos kirim atau bukti bayar sudah terisi | Menekan tombol “Generate PDF” | PDF Faktur terunduh | PDF Faktur terunduh | Valid |
| Mengunduh faktur dengan jenis *Online*  ketika total ongkos kirim atau bukti bayar kosong | Menekan tombol “Generate PDF” kemudian muncul modal untuk mengisikan data total ongkos kirim, bukti bayar, dan memilih *marketplace* kemudian menekan tombol “Save” | PDF Faktur terunduh | PDF Faktur terunduh | Valid |
| Mengunduh faktur dengan jenis *Online*  ketika total ongkos kirim atau bukti bayar sudah terisi | Menekan tombol “Generate PDF” | PDF Faktur terunduh | PDF Faktur terunduh | Valid |
| Mengunduh faktur dengan jenis *Showroom* | Menekan tombol “Generate PDF” | PDF Faktur terunduh | PDF Faktur terunduh | Valid |

Tabel *Test Case* Halaman *Showroom*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menambah faktur dengan jenis *customer* umum | Memilih buku dan jumlah buku kemudian menekan tombol “Submit” | PDF Faktur (*Showroom Receipt*) terunduh | PDF Faktur (*Showroom Receipt*) terunduh | Valid |
| Menambah faktur dengan memilih *customer* dari *database* | *Customer*  Buku | PDF Faktur (*Showroom Receipt*) terunduh | PDF Faktur (*Showroom Receipt*) terunduh | Valid |
| Menambah faktur dengan *customer* baru/belum ada di *database* | Nama *Customer*  Alamat *Customer*  Nomor Telepon *Customer*  Jenis *Customer*  Buku | PDF Faktur (*Showroom Receipt*) terunduh | PDF Faktur (*Showroom Receipt*) terunduh | Valid |
| Menambah faktur dengan jumlah buku yang melebihi stok | Memilih buku dan menekan tombol “Tambah Barang” dengan jumlah buku yang melebihi stok | Muncul peringatan dan buku tidak masuk ke dalam pesanan | Muncul peringatan dan buku tidak masuk ke dalam pesanan | Valid |
| Menambah faktur dengan data yang kurang | Menekan tombol Submit dengan data yang kurang/belum diisi secara keseluruhan | Muncul peringatan di bawah kolom data yang kosong | Muncul peringatan di bawah kolom data yang kosong | Valid |
| Menghapus buku dalam tabel pesanan | Menekan tombol “Hapus” sesuai buku yang ingin dihilangkan dalam tabel pesanan | Buku terhapus dalam tabel pesanan | Buku terhapus dalam tabel pesanan | Valid |
| Menuju halaman *dashboard* faktur | Menekan tulisan “Faktur” pada bagian atas halaman | Menampilkan halaman *dashboard* faktur | Menampilkan halaman *dashboard* faktur | Valid |

Tabel *Test Case* Halaman Pendapatan *Tab* Pendapatan Faktur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan grafik pendapatan seluruh jenis faktur |  | Menampilkan grafik pendapatan seluruh jenis faktur per bulan dalam satu tahun terkini | Menampilkan grafik pendapatan seluruh jenis faktur per bulan dalam satu tahun terkini | Valid |
| Melihat grafik pendapatan faktur pada tahun tertentu | Memilih tahun yang diinginkan kemudian menekan tombol “Filter” | Menampilkan grafik pendapatan seluruh jenis faktur per bulan dalam satu tahun tertentu | Menampilkan grafik pendapatan seluruh jenis faktur per bulan dalam satu tahun tertentu | Valid |
| Melihat grafik pendapatan suatu faktur tertenu | Kredit  Tunai  *Online*  *Showroom* | Menampilkan grafik pendapatan suatu faktur per bulan dalam satu tahun tertentu | Menampilkan grafik pendapatan suatu faktur per bulan dalam satu tahun tertentu | Valid |
| Menampilkan dan me-*reset* data yang ditampilkan seperti awal | Menekan tombol “Reset” | Menampilkan grafik pendapatan seluruh jenis faktur per bulan dalam satu tahun terkini | Menampilkan grafik pendapatan seluruh jenis faktur per bulan dalam satu tahun terkini | Valid |
| Mengunduh Excel laporan pendapatan seluruh jenis faktur pada tahun tertentu | Menekan tombol “Excel” yang berada di sebelah tombol “Filter” | Excel laporan pendapatan terunduh yang berisikan nomor faktur, tanggal dikeluarkan, jenis faktur, status faktur, jumlah pendapatan, dan bukti bayar | Excel laporan pendapatan terunduh yang berisikan nomor faktur, tanggal dikeluarkan, jenis faktur, status faktur, jumlah pendapatan, dan bukti bayar | Valid |
| Melihat tabel detail seluruh pendapatan satu jenis faktur tertentu pada bulan tertentu | Menekan grafik pendapatan faktur tertentu sesuai dengan warnanya | Menampilkan tabel yang berisikan nomor urut, nomor faktur, jenis faktur, tanggal dikeluarkannya faktur, status faktur, dan pendapatan faktur | Menampilkan tabel yang berisikan nomor urut, nomor faktur, jenis faktur, tanggal dikeluarkannya faktur, status faktur, dan pendapatan faktur | Valid |
| Mengunduh Excel laporan pendapatan satu jenis faktur tertentu | Menekan tombol “Excel” yang berada di tabel detail seluruh pendapatan satu jenis faktur | Excel laporan pendapatan satu jenis faktur terunduh yang berisikan nomor faktur, tanggal dikeluarkan, jenis faktur, status faktur, jumlah pendapatan, dan bukti bayar | Excel laporan pendapatan satu jenis faktur terunduh yang berisikan nomor faktur, tanggal dikeluarkan, jenis faktur, status faktur, jumlah pendapatan, dan bukti bayar | Valid |

Tabel *Test Case* Halaman Pendapatan *Tab* Detail Pendapatan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan perbandingan pendapatan antar jenis faktur |  | Menampilkan grafik dalam bentuk *pie chart* total pendapatan masing-masing jenis faktur dalam satu tahun terkini dan tabel jumlah pendapatan masing-masing jenis faktur | Menampilkan grafik dalam bentuk *pie chart* total pendapatan masing-masing jenis faktur dalam satu tahun terkini tabel jumlah pendapatan masing-masing jenis faktur | Valid |
| Menampilkan perbandingan pendapatan antar jenis faktur pada tahun tertentu | Memilih tahun yang diinginkan kemudian menekan tombol “Filter” | Menampilkan grafik dalam bentuk *pie chart* total pendapatan masing-masing jenis faktur dalam tahun yang dipilih | Menampilkan grafik dalam bentuk *pie chart* total pendapatan masing-masing jenis faktur dalam tahun yang dipilih | Valid |
| Menampilkan perbandingan pendapatan antar jenis faktur dalam satu periode tertentu | Memilih periode waktu yaitu satu tahun atau bulan tertentu kemudian menekan tombol “Filter” | Menampilkan grafik dalam bentuk *pie chart* total pendapatan masing-masing jenis faktur dalam satu periode tertentu | Menampilkan grafik dalam bentuk *pie chart* total pendapatan masing-masing jenis faktur dalam satu periode tertentu | Valid |
| Menampilkan dan me-*reset* data yang ditampilkan seperti awal | Menekan tombol “Reset” | Menampilkan grafik dalam bentuk *pie chart* total pendapatan masing-masing jenis faktur dalam satu tahun terkini dan tabel jumlah pendapatan masing-masing jenis faktur | Menampilkan grafik dalam bentuk *pie chart* total pendapatan masing-masing jenis faktur dalam satu tahun terkini dan tabel jumlah pendapatan masing-masing jenis faktur | Valid |
| Mengunduh Excel laporan pendapatan seluruh jenis faktur pada tahun tertentu | Menekan tombol “Excel” | Excel laporan pendapatan terunduh | Excel laporan pendapatan terunduh | Valid |

Tabel *Test Case* Halaman Royalti *Tab* Tagihan Royalti

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan daftar penerima royalti |  | Menampilkan tabel daftar penerima royalti yang berisikan nomor urut, nama penerima royalti, periode pembayaran royalti, jumlah royalti, dan status pembayaran royalti dengan periode hingga H-1 waktu terkini | Menampilkan tabel daftar penerima royalti yang berisikan nomor urut, nama penerima royalti, periode pembayaran royalti, jumlah royalti, dan status pembayaran royalti dengan periode hingga H-1 waktu terkini | Valid |
| Mengubah jumlah data yang ditampilkan pada halaman *dashboard* | 10  25  50  100 | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Valid |
| Melakukan pencarian data dengan tanggal akhir periode penerimaan royalti pada satu waktu tertentu | Tanggal terakhir periode | Menampilkan tabel daftar penerima royalti hingga periode yang telah dipilih | Menampilkan tabel daftar penerima royalti hingga periode yang telah dipilih | Valid |
| Melakukan pencarian data berdasarkan nama penulis | Mencari nama penulis di “Filter Penulis” | Menunjukkan data royalti berdasarkan nama penulis yang diinginkan | Menunjukkan data royalti berdasarkan nama penulis yang diinginkan | Valid |
| Menampilkan dan me-*reset* data yang ditampilkan seperti awal | Menekan tombol “Reset” | Menampilkan tabel daftar penerima royalti yang berisikan nomor urut, nama penerima royalti, jumlah penjualan buku, jumlah royalti, periode pembayaran royalti, dan status pembayaran royalti dengan periode hingga H-1 waktu terkini | Menampilkan tabel daftar penerima royalti yang berisikan nomor urut, nama penerima royalti, jumlah penjualan buku, jumlah royalti, periode pembayaran royalti, dan status pembayaran royalti dengan periode hingga H-1 waktu terkini | Valid |
| Menuju ke halaman detail penulis tertentu | Menekan nama penerima royalti | Menuju ke halaman detail penulis *tab* “Riwayat Royalti” | Menuju ke halaman detail penulis *tab* “Riwayat Royalti” | Valid |
| Menuju ke halaman pengajuan dan pembayaran royalti penulis | Menekan tombol “Form Pengajuan” | Menuju ke halaman pengajuan dan pembayaran royalti penulis | Menuju ke halaman pengajuan dan pembayaran royalti penulis | Valid |

Tabel *Test Case* Halaman Pengajuan dan Pembayaran Royalti Penulis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Membuka halaman ketika tanggal awal atau tanggal akhir periode belum dimasukkan |  | Menampilkan peringatan untuk memasukkan tanggal awal dan tanggal akhir periode | Menampilkan peringatan untuk memasukkan tanggal awal dan tanggal akhir periode | Valid |
| Memasukkan tanggal awal dan tanggal akhir periode | Tanggal awal dan tanggal akhir periode kemudian menekan tombol “Filter” | Menampilkan tabel pengajuan royalti penulis yang berisikan nomor urut, judul buku, jumlah buku terjual, hasil penjualan, dan total royalti yang didapatkan dalam rentang periode yang dimasukkan | Menampilkan tabel pengajuan royalti penulis yang berisikan nomor urut, judul buku, jumlah buku terjual, hasil penjualan, dan total royalti yang didapatkan dalam rentang periode yang dimasukkan | Valid |
| Mengajukan pembayaran royalti penulis | Menekan tombol “Ajukan Royalti” kemudian tekan tombol “Confirm” pada modal | Muncul peringatan sukses berupa *toast* dan menuju ke halaman royalti *tab* tagihan royalti serta status royalti penulis berubah menjadi “Diajukan” | Muncul peringatan sukses berupa *toast* dan menuju ke halaman royalti *tab* tagihan royalti serta status royalti penulis berubah menjadi “Diajukan” | Valid |
| Membayar royalti ketika sudah mengajukan pembayaran royalti penulis sebelumnya | Menekan tombol “Bayar” kemudian memasukkan bukti bayar pada modal dan menekan tombol “Confirm” | Menuju ke halaman *dashboard* royalti | Menuju ke halaman *dashboard* royalti | Valid |
| Menuju ke halaman detail riwayat royalti suatu penulis | Menekan tombol “Detail” | Menuju ke halaman detail riwayat royalti suatu penulis | Menuju ke halaman detail riwayat royalti suatu penulis | Valid |
| Menuju ke halaman penerimaan royalti ketika status royalti penulis masih “Diajukan” |  | Muncul tabel detail royalti yang belum dibayar yang berisikan tanggal mulai periode, tanggal akhir periode, dan total royalti | Muncul tabel detail royalti yang belum dibayar yang berisikan tanggal mulai periode, tanggal akhir periode, dan total royalti | Valid |

Tabel *Test Case* Halaman Royalti *Tab* Riwayat Royalti

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan riwayat royalti semua penulis |  | Menampilkan tabel yang berisi nomor urut, nama penulis, periode royalti, jumlah royalti, status royalti, dan tanggal royalti dibayarkan | Menampilkan tabel yang berisi nomor urut, nama penulis, periode royalti, jumlah royalti, status royalti, dan tanggal royalti dibayarkan | Valid |
| Mengubah jumlah data yang ditampilkan pada halaman *dashboard* | 10  25  50  100 | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Valid |
| Melakukan pencarian data dengan tanggal awal periode penerimaan royalti pada satu waktu tertentu | Tanggal awal periode | Menampilkan tabel daftar penerima royalti dari periode yang telah dipilih | Menampilkan tabel daftar penerima royalti dari periode yang telah dipilih | Valid |
| Melakukan pencarian data dengan tanggal akhir periode penerimaan royalti pada satu waktu tertentu | Tanggal terakhir periode | Menampilkan tabel daftar penerima royalti hingga periode yang telah dipilih | Menampilkan tabel daftar penerima royalti hingga periode yang telah dipilih | Valid |
| Melakukan pencarian data berdasarkan nama penulis | Mencari nama penulis di “Filter Penulis” | Menunjukkan data royalti berdasarkan nama penulis yang diinginkan | Menunjukkan data royalti berdasarkan nama penulis yang diinginkan | Valid |
| Menampilkan dan me-*reset* data yang ditampilkan seperti awal | Menekan tombol “Reset” | Menampilkan tabel yang berisi nomor urut, nama penulis, tanggal mulai periode royalti, tanggal akhir periode royalti, tanggal royalti dibayarkan, dan total royalti | Menampilkan tabel yang berisi nomor urut, nama penulis, tanggal mulai periode royalti, tanggal akhir periode royalti, tanggal royalti dibayarkan, dan total royalti | Valid |
| Menuju ke halaman detail penulis tertentu | Menekan nama penerima royalti | Menuju ke halaman detail penulis *tab* “Riwayat Royalti” | Menuju ke halaman detail penulis *tab* “Riwayat Royalti” | Valid |
| Menuju ke halaman detail riwayat royalti suatu penulis | Menekan tombol “Detail” atau nama penulis tertentu | Menuju ke halaman detail riwayat royalti suatu penulis | Menuju ke halaman detail riwayat royalti suatu penulis | Valid |

Tabel *Test Case* Halaman Detail Penerimaan Royalti Penulis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan tabel detail penerimaan royalti penulis tertentu |  | Menampilkan halaman yang berisikan nama penulis, periode royalti, status royalti, bukti bayar, serta tabel yang berisikan nomor urut, judul buku yang terjual, jumlah buku yang terjual, jumlah penjualan, dan besarnya royalti | Menampilkan halaman yang berisikan nama penulis, periode royalti, status royalti, bukti bayar, serta tabel yang berisikan nomor urut, judul buku yang terjual, jumlah buku yang terjual, jumlah penjualan, dan besarnya royalti | Valid |
| Membayar royalti ketika sudah mengajukan pembayaran royalti penulis sebelumnya | Menekan tombol “Bayar” kemudian memasukkan bukti bayar pada modal dan menekan tombol “Confirm” | Menuju ke halaman royalti *tab* tagihan royalti | Menuju ke halaman royalti *tab* tagihan royalti | Valid |
| Mengunduh laporan royalti penulis tertentu | Menekan tombol “Generate PDF” | PDF laporan royalti penulis tersebut terunduh | PDF laporan royalti penulis tersebut terunduh | Valid |
| Menuju ke halaman *dashboard* royalti | Menekan tombol “Kembali” | Menuju ke halaman pengajuan dan pembayaran royalti penulis | Menuju ke halaman pengajuan dan pembayaran royalti penulis | Valid |

Tabel *Test Case* Halaman *Customer*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Skenario | Masukkan | Keluaran yang diharapkan | Keluaran sebenarnya | Hasil |
| Menampilkan identitas *customer* yang ada di *database* |  | Menampilkan tabel yang berisikan nomor urut, nama *customer*, alamat *customer*, nomor telepon, *email customer*, dan jenis *customer* | Menampilkan tabel yang berisikan nomor urut, nama *customer*, alamat *customer*, nomor telepon, *email customer*, dan jenis *customer* | Valid |
| Mengubah jumlah data yang ditampilkan pada halaman *customer* | 10  25  50  100 | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Menunjukkan jumlah data pada halaman sesuai dengan masukkan | Valid |
| Menyortir dan menampilkan jenis *customer* tertentu | Distributor  Reseller  Penulis  Member  Umum | Menunjukkan data yang dipilih sesuai dengan masukkan | Menunjukkan data yang dipilih sesuai dengan masukkan | Valid |
| Melakukan pencarian data sesuai dengan masukkan pengguna | Data berbentuk teks | Menunjukkan data sesuai dengan masukkan | Menunjukkan data sesuai dengan masukkan | Valid |
| Menampilkan dan me-*reset* data yang ditampilkan seperti awal | Menekan tombol “Reset” | Menampilkan tabel yang berisikan nomor urut, nama *customer*, alamat *customer*, nomor telepon, *email customer*, dan jenis *customer* | Menampilkan tabel yang berisikan nomor urut, nama *customer*, alamat *customer*, nomor telepon, *email customer*, dan jenis *customer* | Valid |
| Menambah *customer* baru | Menekan tombol “Tambah” kemudian isi data pada modal dan menekan tombol “Save” | *Customer* yang dimasukkan akan tersimpan dan ditampilkan pada halaman *customer* | *Customer* yang dimasukkan akan tersimpan dan ditampilkan pada halaman *customer* | Valid |
| Menambah *customer* baru dengan data yang tidak lengkap | Menekan tombol “Tambah” lalu menekan tombol “Save” dengan data yang tidak lengkap | Muncul peringatan di bawah kolom data yang kosong | Muncul peringatan di bawah kolom data yang kosong | Valid |
| Mengubah diskon buku berdasarkan jenis *customer* | Menekan tombol “Diskon” | Muncul modal yang berisikan diskon masing-masing jenis *customer* | Muncul modal yang berisikan diskon masing-masing jenis *customer* | Valid |
| Mengubah data suatu *customer* | Menekan tombol dengan *icon* bergambar pensil | Muncul modal yang berisikan data *customer* yang akan diubah | Muncul modal yang berisikan data *customer* yang akan diubah | Valid |
| Menghapus *customer* | Menekan tombol dengan *icon* bergambar tempat sampah | Muncul modal peringatan hapus *customer* | Muncul modal peringatan hapus *customer* | Valid |

**3.2.3 *User Experience***

Partisipan dari *Usability Testing* adalah calon pengguna sistem pemasaran SIGAP yaitu karyawan bagian pemasaran UGM Press. Pengujian dilakukan secara langsung bertempat di UGM Press, Jl. Sendok, Karanggayam, Sleman, Yogyakarta. Pelaksanaan pengujian diawali dengan pembacaan skenario oleh penulis dan menjelaskan skenario kepada partisipan untuk melakukan aktivitas sesuai dengan skenario. Setelah partisipan memahami skenario, kemudian partisipan melakukan pengujian dan diberi kesempatan untuk mengajukan pertanyaan seputar antarmuka sistem. Kemudian partisipan mengisi kuesioner SUS (*System Usability Scale*) dan menyerahkan hasilnya kepada penulis. Terkait skenario yang harus dilakukan partisipan, berikut merupakan skenario pengujian yang dilakukan oleh partisipan.

1. Bapak/Ibu ingin membuat faktur dengan jenis faktur yang berbeda-beda.
2. Bapak/Ibu ingin mengunduh faktur untuk dikirimkan kepada pelanggan UGM Press.
3. Bapak/Ibu ingin melihat laporan pendapatan yang didapatkan pada waktu tertentu.
4. Bapak/Ibu ingin mengajukan dan melakukan pembayaran royalti terhadap suatu penulis pada periode tertentu.

Daftar kuesioner dalam SUS (*System Usability Scale*) adalah sebagai berikut.

1. Saya merasa saya akan menggunakan Sistem Informasi ini ke depannya
2. Saya menemukan fungsi yang tidak penting di Sistem Informasi ini
3. Saya merasa Sistem Informasi ini mudah digunakan
4. Saya merasa membutuhkan bantuan dari technical support untuk menggunakan Sistem Informasi ini
5. Saya menemukan Sistem Informasi ini yang berjalan dengan baik
6. Saya merasa banyak inkonsistensi pada Sistem Informasi ini
7. Saya merasa orang-orang akan mudah dalam menggunakan Sistem Informasi ini
8. Saya merasa Sistem Informasi ini rumit
9. Saya merasa percaya diri dalam menggunakan Sistem Informasi ini
10. Saya perlu belajar banyak hal sebelum bisa menggunakan Sistem Informasi ini

SUS memiliki 10 pertanyaan dan 5 pilihan jawaban yang terdiri dari sangat tidak setuju, tidak setuju, ragu-ragu, setuju, dan sangat setuju. Skor dari masing-masing pertanyaan berada pada skala nilai 1 sampai dengan 5. Berikut merupakan skor nilai pengujian menggunakan metode SUS dari masing-masing partisipan.

Tabel Skor Nilai Pengujian menggunakan Metode SUS

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Responden | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
| Irwan Ruslan | 5 | 1 | 5 | 2 | 5 | 1 | 5 | 1 | 5 | 2 |
| Roselli N. S. | 5 | 1 | 5 | 5 | 5 | 1 | 5 | 1 | 5 | 4 |
| Febriarto Wahyu Aji | 5 | 3 | 5 | 2 | 4 | 3 | 5 | 2 | 4 | 2 |

**3.2.4. Evaluasi Sistem Pemasaran SIGAP**

Setelah dilakukan pengujian melalui beberapa skenario yang diberikan, dilakukan perbandingan antara permasalahan utama yang didefinisikan sebelumnya dan sistem informasi yang dikembangkan dengan menghadirkan fitur untuk membantu menyelesaikan permasalahan tersebut. Berikut merupakan perbandingan antara proses bisnis sebelum dan sesudah menggunakan sistem pemasaran SIGAP.

1. Pembuatan faktur sebagai bukti pemesanan dan pembayaran buku di UGM Press

* Kondisi saat ini : Pembuatan faktur tidak terintegrasi secara *real time* dengan jumlah stok barang
* Menggunakan SIGAP : Pembuatan faktur terintegrasi dengan jumlah stok barang yang ada di UGM Press secara *real time* dan seluruh faktur dapat dilihat secara keseluruhan pada sistem

1. Penghitungan pendapatan UGM Press pada satu periode tertentu

* Kondisi saat ini : Penghitungan manual pada akhir periode menggunakan Microsoft Excel
* Menggunakan SIGAP : Penghitungan otomatis oleh SIGAP dan dapat dilihat berdasarkan satu periode tertentu secara fleksibel

1. Melakukan pengajuan dan pembayaran royalti kepada penulis

* Kondisi saat ini : Penghitungan royalti masih manual setiap 6 bulan sekali menggunakan Microsoft Excel
* Menggunakan SIGAP : Penghitungan royalti secara otomatis serta waktu pengajuan dan pembayaran royalti dapat dilakukan secara fleksibel

**3.3. Analisis Pengujian**

**3.3.1. Analisis Pengujian *Back*-*End***

**3.3.2. Analisis Pengujian *Front*-*End***

Berdasarkan hasil pengujian *front*-*end* menggunakan metode *blackbox testing*, semua hasil yang dihasilkan oleh sistem sesuai dengan hasil yang diharapkan dan berjalan dengan baik sesuai dengan fungsinya masing-masing. Tampilan sudah dikembangkan sesuai dengan fungsi yang diinginkan oleh pengguna sistem sehingga sistem dapat digunakan sesuai kebutuhan pengguna.

**3.3.3. Analisis Pengujian *User Experience***

Setelah mendapatkan skor nilai pengujian masing-masing partisipan, kemudian dilakukan perhitungan nilai berdasarkan aturan perhitungan SUS. Dalam melakukan perhitungan, terdapat beberapa aturan sebagai berikut :

1. Setiap pertanyaan bernomor ganjil, skor setiap pertanyaan yang didapat dari skor pengguna akan dikurangi 1
2. Setiap pertanyaan bernomor genap, skor akhir didapat dari nilai 5 dikurangi skor pertanyaan yang didapat dari pengguna
3. Skor SUS didapat dari hasil penjumlahan skor setiap pertanyaan yang kemudian dikali 2,5

Berikut merupakan hasil perhitungan skor masing-masing partisipan mengikuti aturan SUS di atas.

Tabel Analisis Skor Nilai Pengujian menggunakan Metode SUS

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Responden | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Jml | Skor SUS |
| Irwan Ruslan | 4 | 4 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 3 | 38 | 95 |
| Roselli N. S. | 4 | 4 | 4 | 0 | 4 | 4 | 4 | 4 | 4 | 1 | 33 | 82,5 |
| Febriarto Wahyu Aji | 4 | 2 | 4 | 3 | 1 | 2 | 0 | 3 | 1 | 3 | 23 | 57,5 |
| **Skor SUS rata-rata** | | | | | | | | | | | | **78,3** |

Untuk mencari skor SUS rata-rata sistem pemasaran SIGAP oleh partisipan, dilakukan perhitungan rata-rata dengan cara sebagai berikut.

(1)

Persamaan (1) adalah persamaan mencari nilai rata-rata skor SUS

Skor SUS rata-rata () didapat dari jumlah skor SUS seluruh partisipan () dibagi dengan jumlah partisipan (). Dengan mengetahui skor SUS rata-rata, dapat disimpulkan bahwa sistem pemasaran SIGAP mendapatkan nilai **B** yang mengacu pada tabel penilaian sebagai berikut.

Tabel Penilaian SUS

|  |  |  |
| --- | --- | --- |
| **SUS *Score*** | ***Grade*** | ***Adjective Rating*** |
| > 80,3 | A | *Excellent* |
| 68 - 80,3 | B | *Good* |
| 68 | C | *Okay* |
| 51 - 68 | D | *Poor* |
| < 51 | F | *Awful* |

**3.3.4. Analisis Evaluasi Sistem Pemasaran SIGAP**

Setelah membandingkan antara proses bisnis pemasaran UGM Press sebelum dan setelah menggunakan SIGAP dapat disimpulkan bahwa permasalahan utama yang didefinisikan pada dokumen C250 sudah teratasi. Pembuatan faktur sebelum menggunakan SIGAP, jumlah buku tidak terintegrasi secara aktual dengan jumlah buku yang ada di stok UGM Press saat itu, sehingga menimbulkan kegagalan pemrosesan pesanan atau pesanan diproses dalam waktu yang cukup lama menunggu ketersediaan stok barang. Sistem pemasaran SIGAP menyediakan fitur pembuatan faktur yang terintegrasi secara *real time* dengan jumlah stok buku terkini sehingga permasalahan data yang tidak aktual dapat terselesaikan dan pemrosesan pesanan dapat dilakukan tanpa hambatan guna menaikkan kepuasan pelanggan terhadap pelayanan UGM Press.

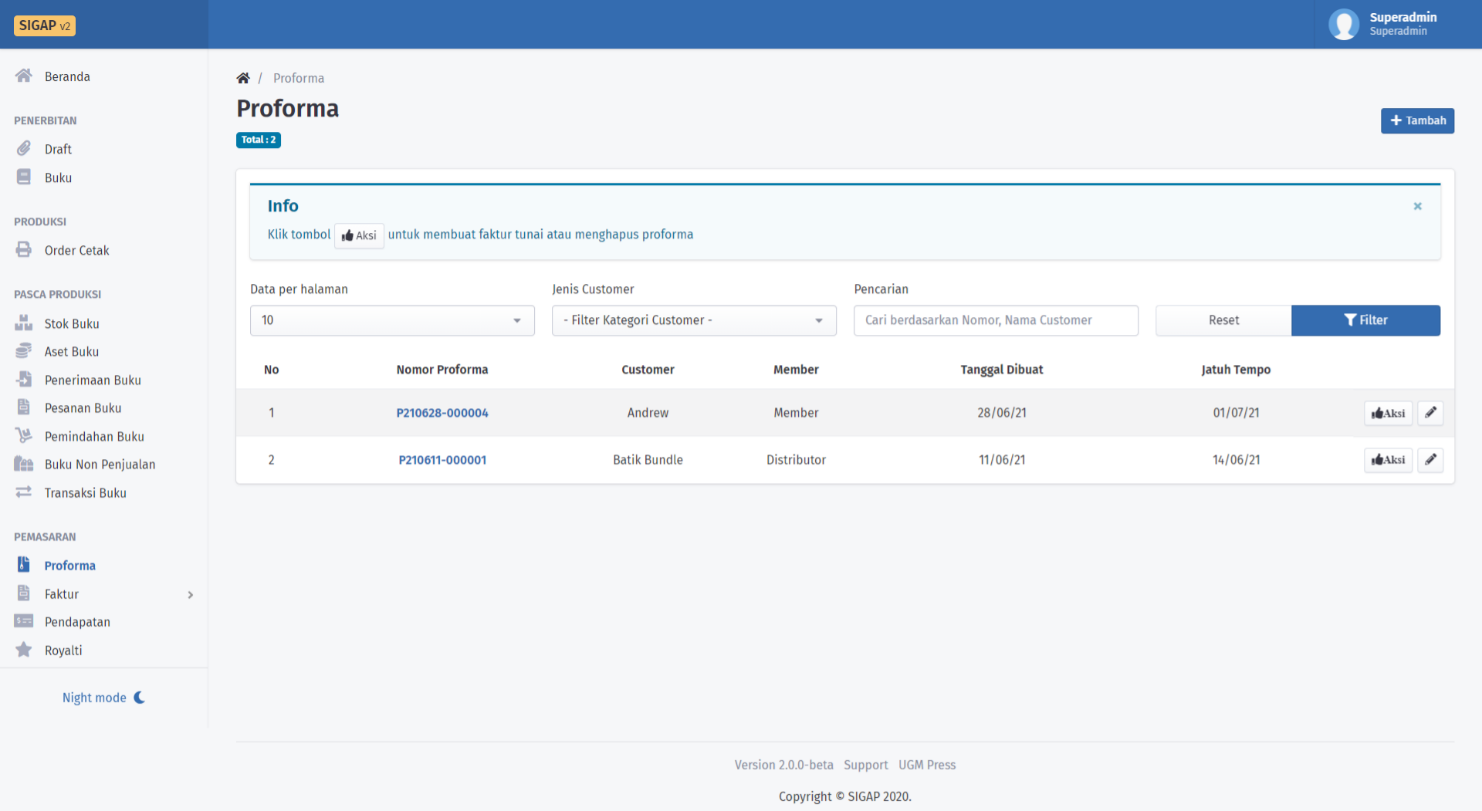
Penghitungan jumlah pendapatan sebelum menggunakan SIGAP masih dilakukan secara manual dengan memasukkan data faktur ke Microsoft Excel kemudian dilakukan penghitungan setelah proses *input* data selesai. Proses tersebut membutuhkan cukup banyak waktu karena jumlah faktur yang ada dikumpulkan dari awal periode dan jumlahnya menumpuk, sehingga memungkinkan adanya data yang terlewat dan tidak sesuai. Menggunakan SIGAP, penghitungan jumlah pendapatan dilakukan secara otomatis oleh sistem sehingga pengguna dapat melihat rekap data tersebut kapan saja dan laporan pendapatan dapat ditampilkan serta diunduh berdasarkan satu periode tertentu sesuai dengan masukkan pengguna.

Penghitungan royalti merupakan salah satu pekerjaan bagian pemasaran UGM Press dan berhubungan dengan *stakeholder* UGM Press yaitu penulis buku. Penghitungan royalti sebelum menggunakan SIGAP, prosesnya dilakukan secara manual setiap 6 bulan sekali dan penghitungan dilakukan menggunakan Microsoft Excel dengan memasukkan data jumlah pendapatan faktur dan memilahnya berdasarkan masing-masing judul buku yang terjual kemudian mengelompokkan jumlah pendapatan tersebut dan menghitung royalti berdasarkan penulisnya. Proses tersebut rentan terhadap data yang terlewat dan tidak sesuai dengan kondisi sebenarnya, maka dari itu sistem pemasaran SIGAP menghadirkan fitur untuk mengajukan dan membayar royalti dengan data-data yang sudah terintegrasi pada sistem. Jumlah royalti penulis yang belum terbayar akan ditampilkan pada satu halaman sehingga memudahkan pengguna untuk melakukan pengecekan dan menghindari pembayaran royalti yang tidak sesuai. Sistem pemasaran SIGAP memungkinkan pengguna untuk mengajukan dan membayar royalti kapanpun tanpa harus menunggu setiap 6 bulan sekali. Hal ini menyebabkan peningkatan kepercayaan dan kepuasan penulis terhadap pelayanan UGM Press karena proses pencairan royalti dapat disesuaikan sesuai dengan kemauan penulis.

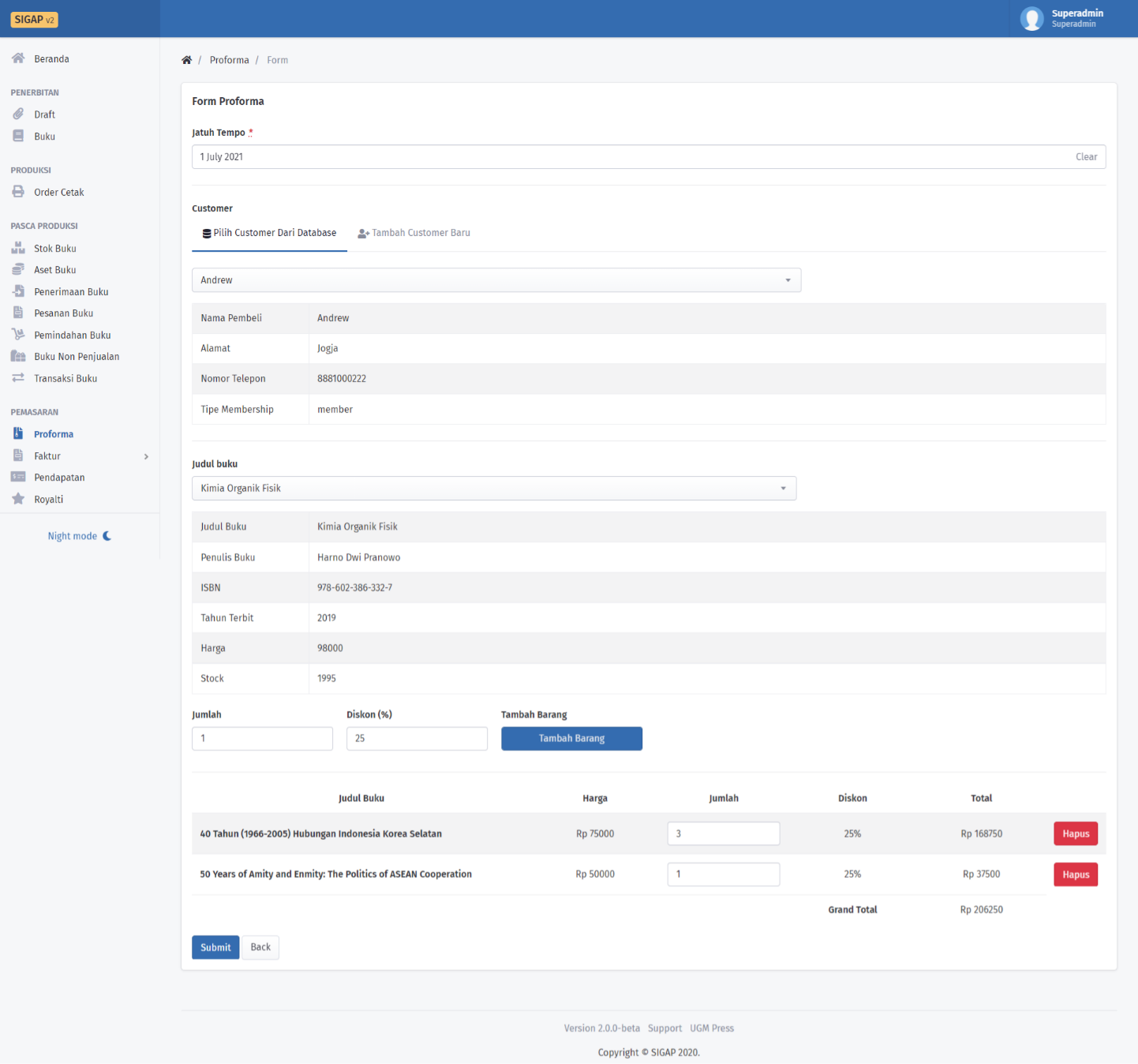
# KESIMPULAN

# REFERENSI

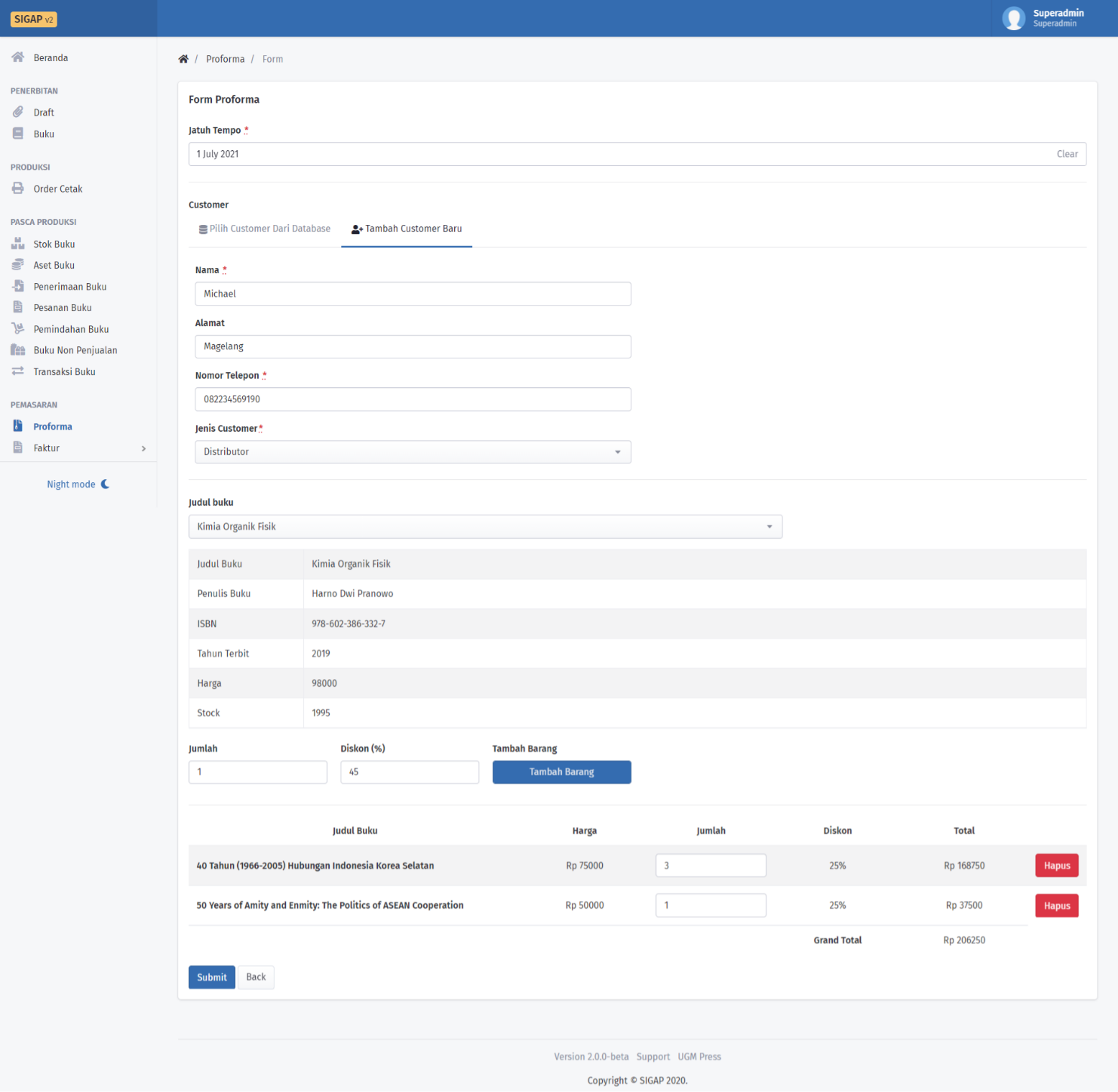
# LAMPIRAN-LAMPIRAN



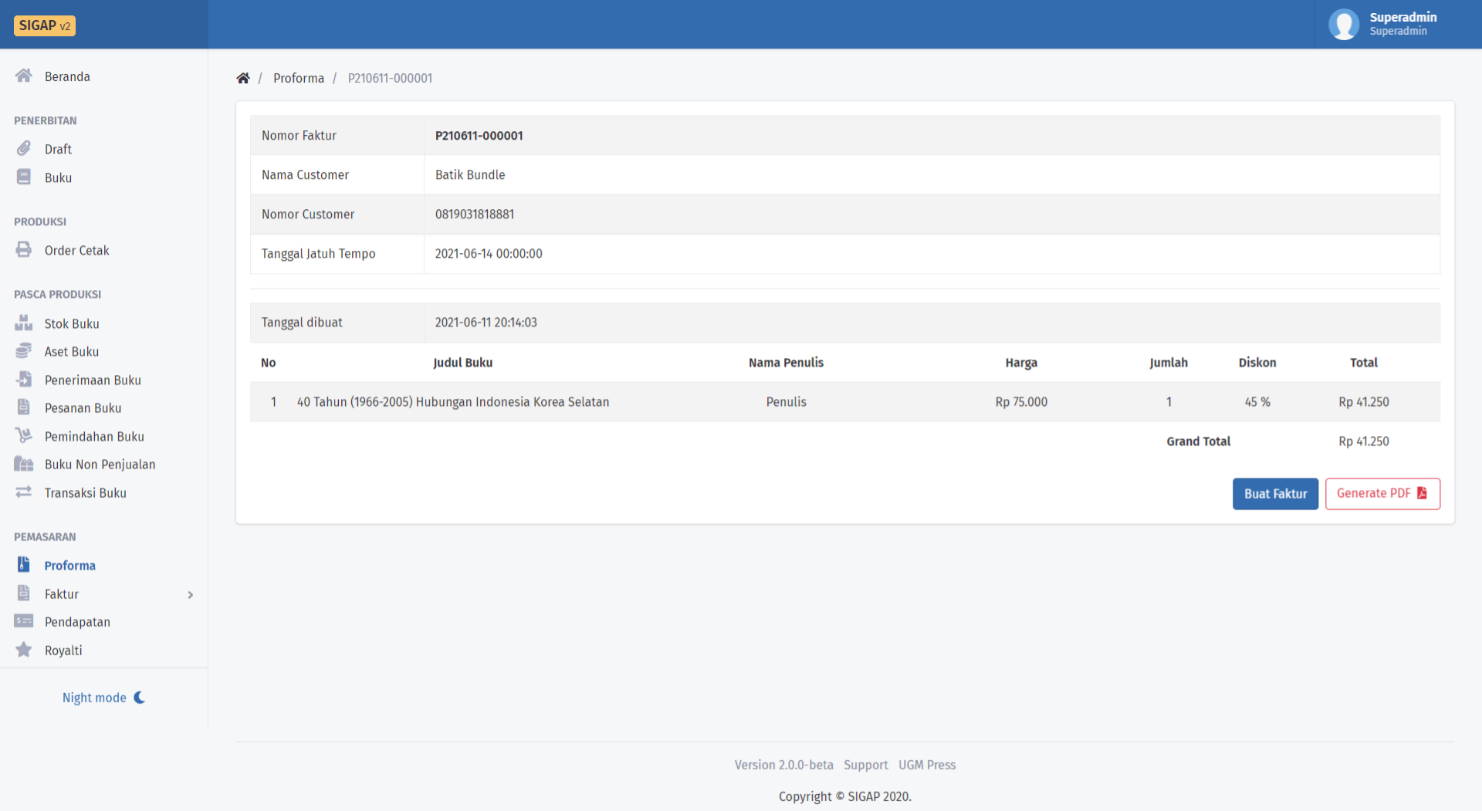
Gambar 1. Halaman Proforma



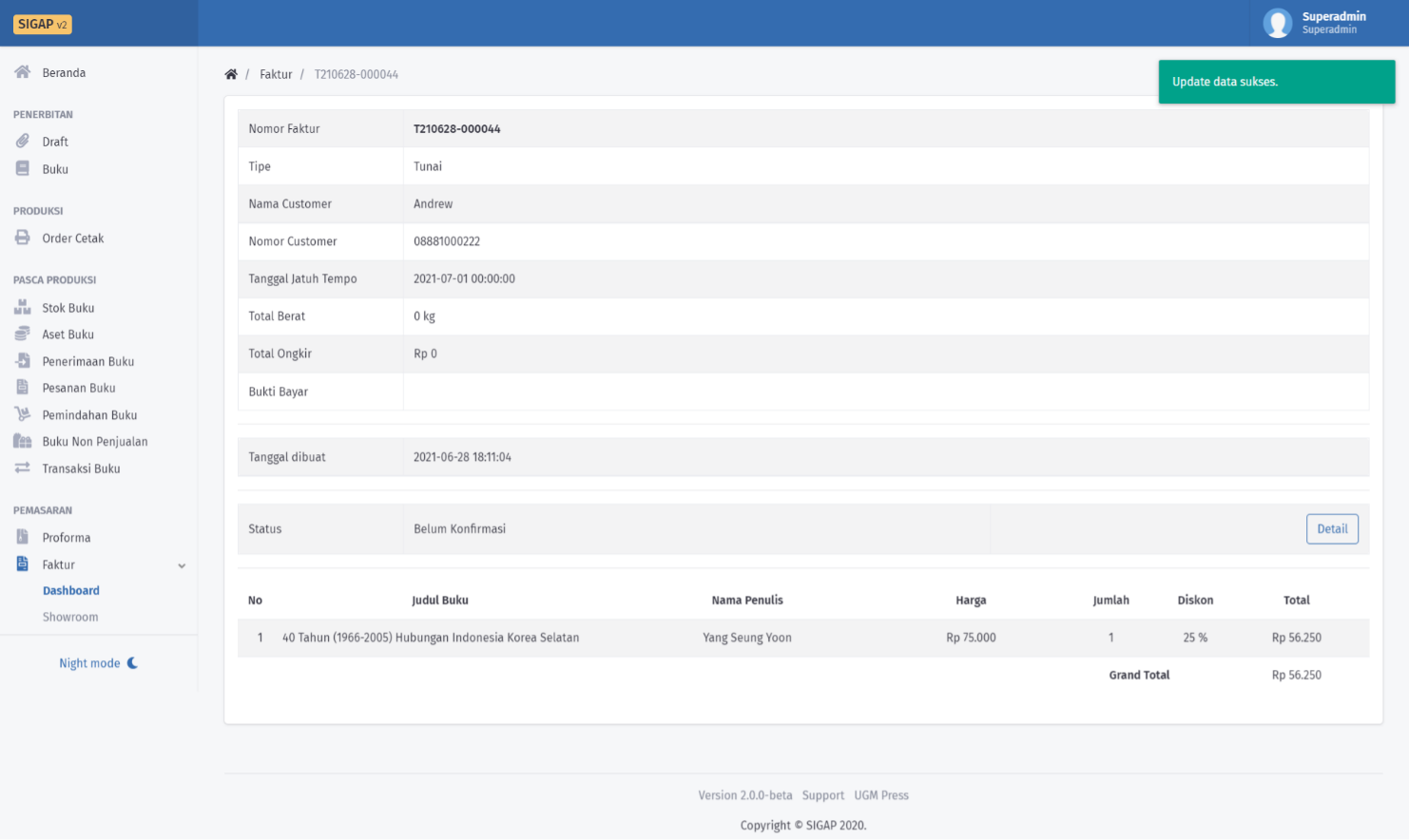
Gambar 2. Halaman Tambah Proforma jika Memilih *Customer* dari *Database*



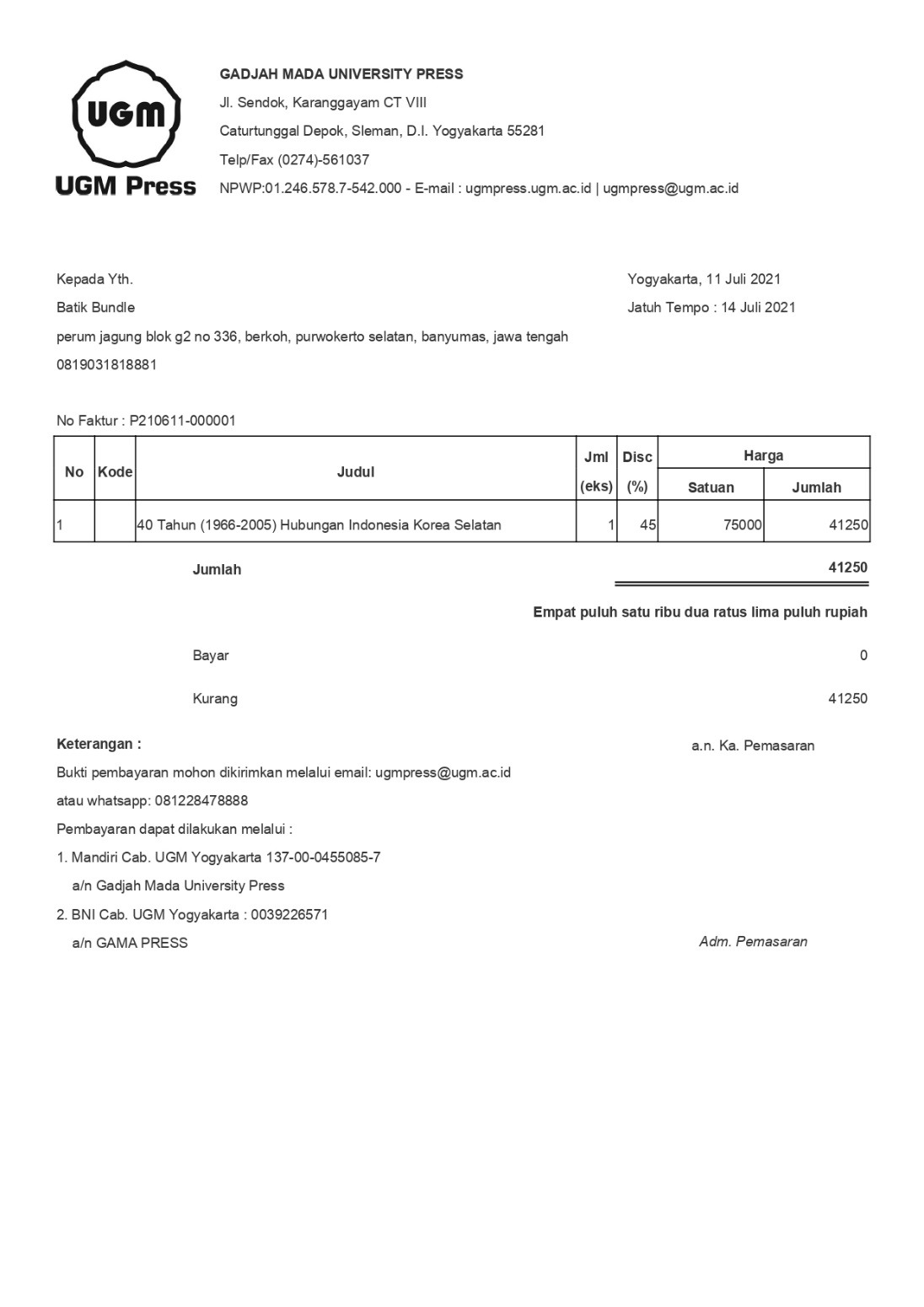
Gambar 3. Halaman Tambah Proforma jika ingin Memasukkan dan Memilih *Customer* Baru



Gambar 4. Halaman Detail Suatu Proforma



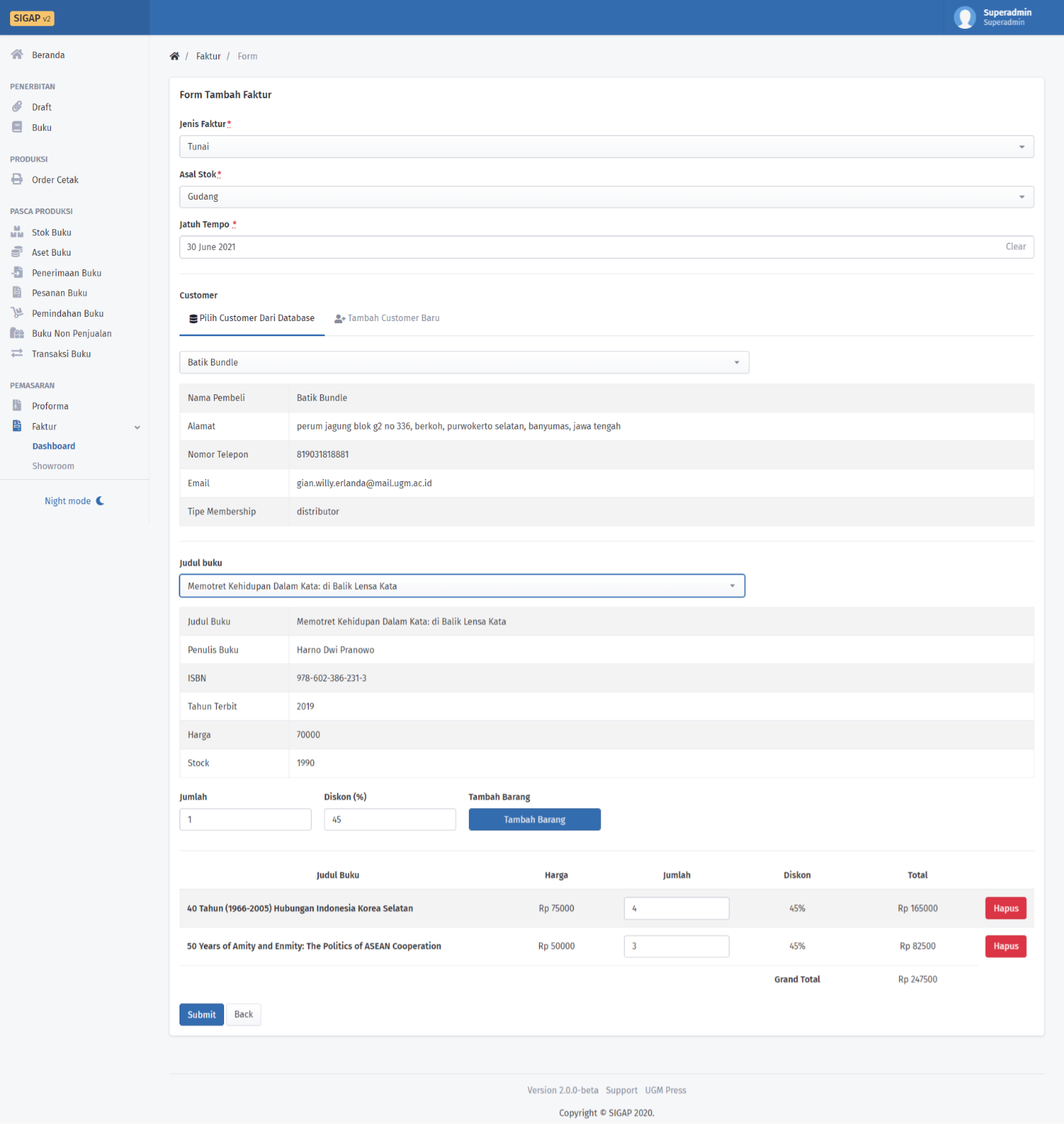
Gambar 5. Halaman ketika Proforma diubah menjadi Faktur Tunai



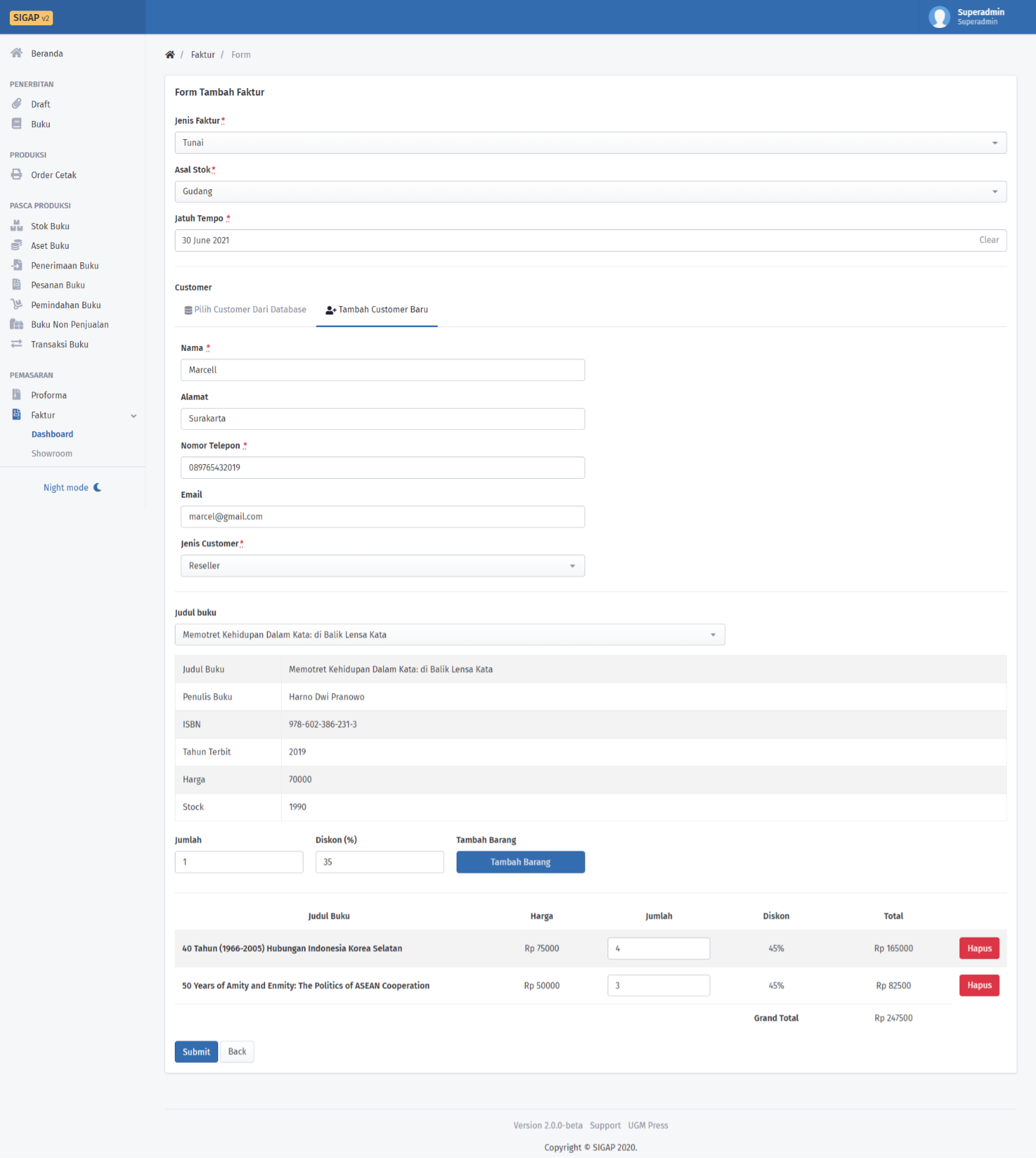
Gambar 6. Tampilan PDF Proforma



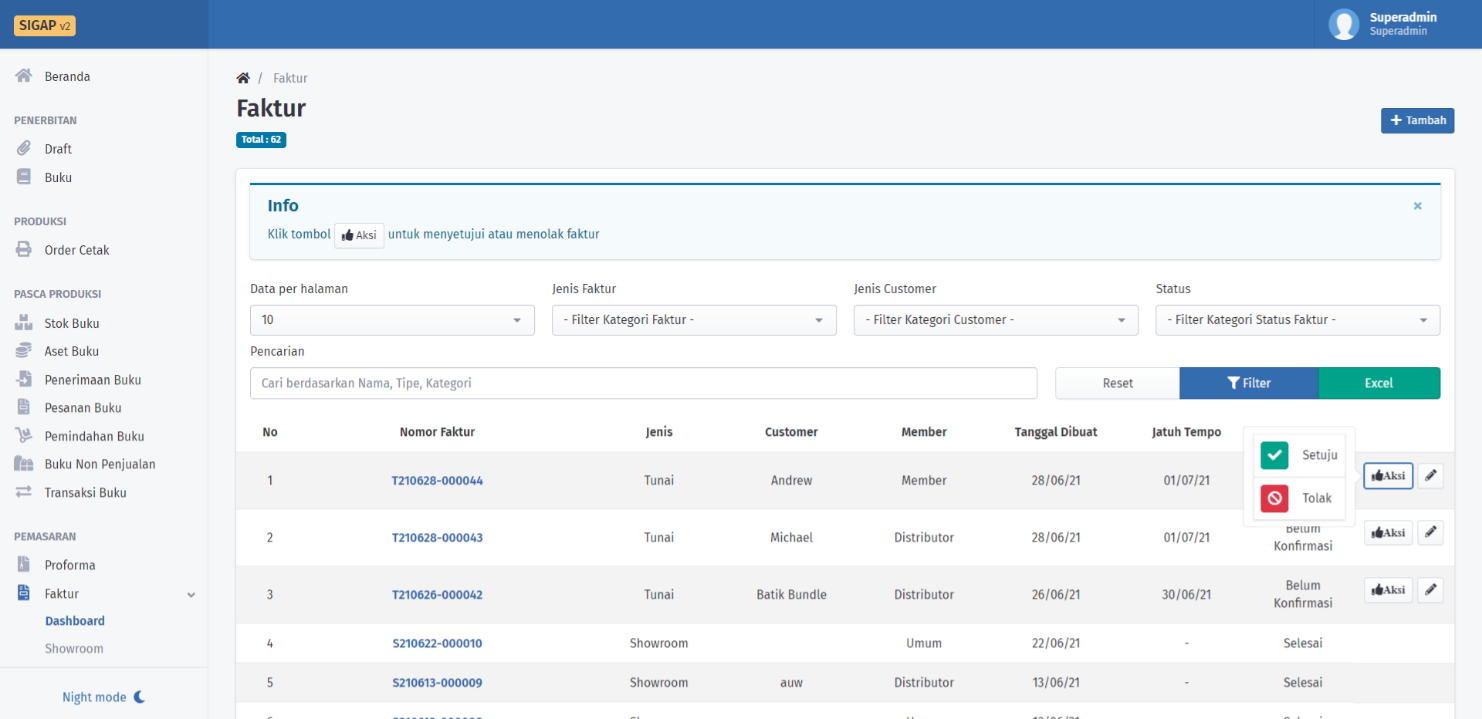
Gambar 7. Halaman *Dashboard* Faktur



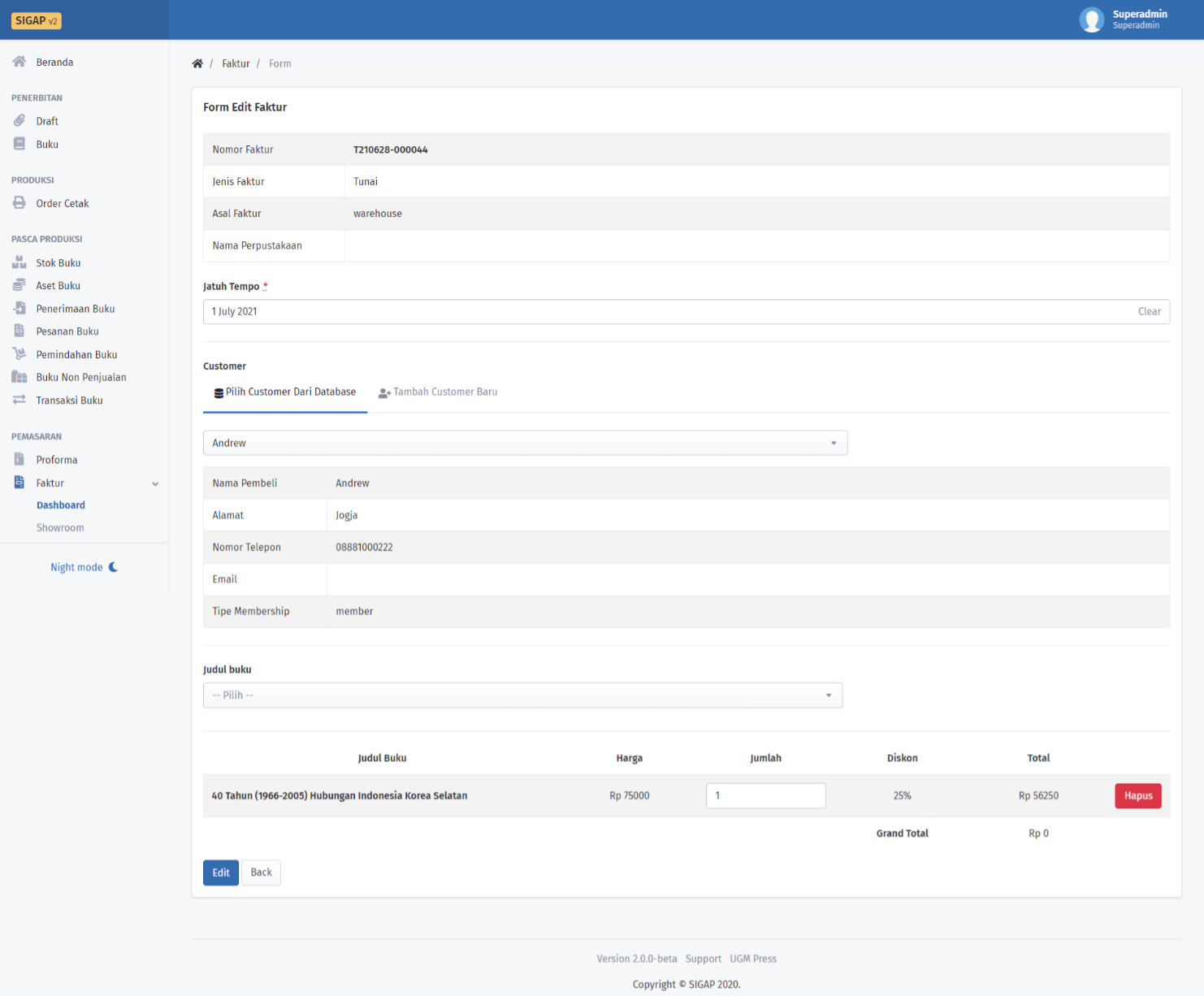
Gambar 8. Halaman Tambah Faktur Ketika Memilih *Customer* dari *Database*



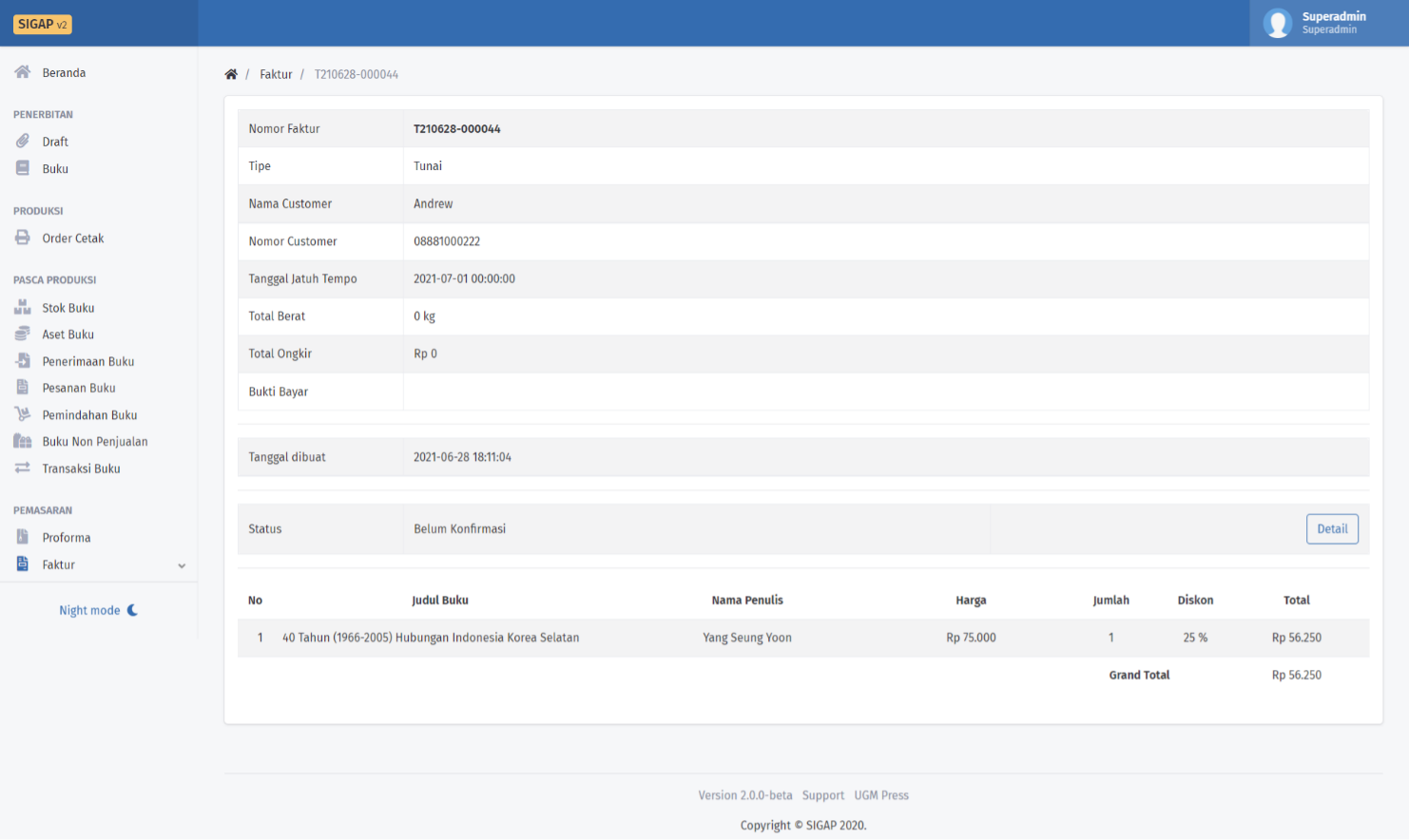
Gambar 9. Halaman Tambah Faktur jika ingin Memasukkan dan Memilih *Customer* Baru



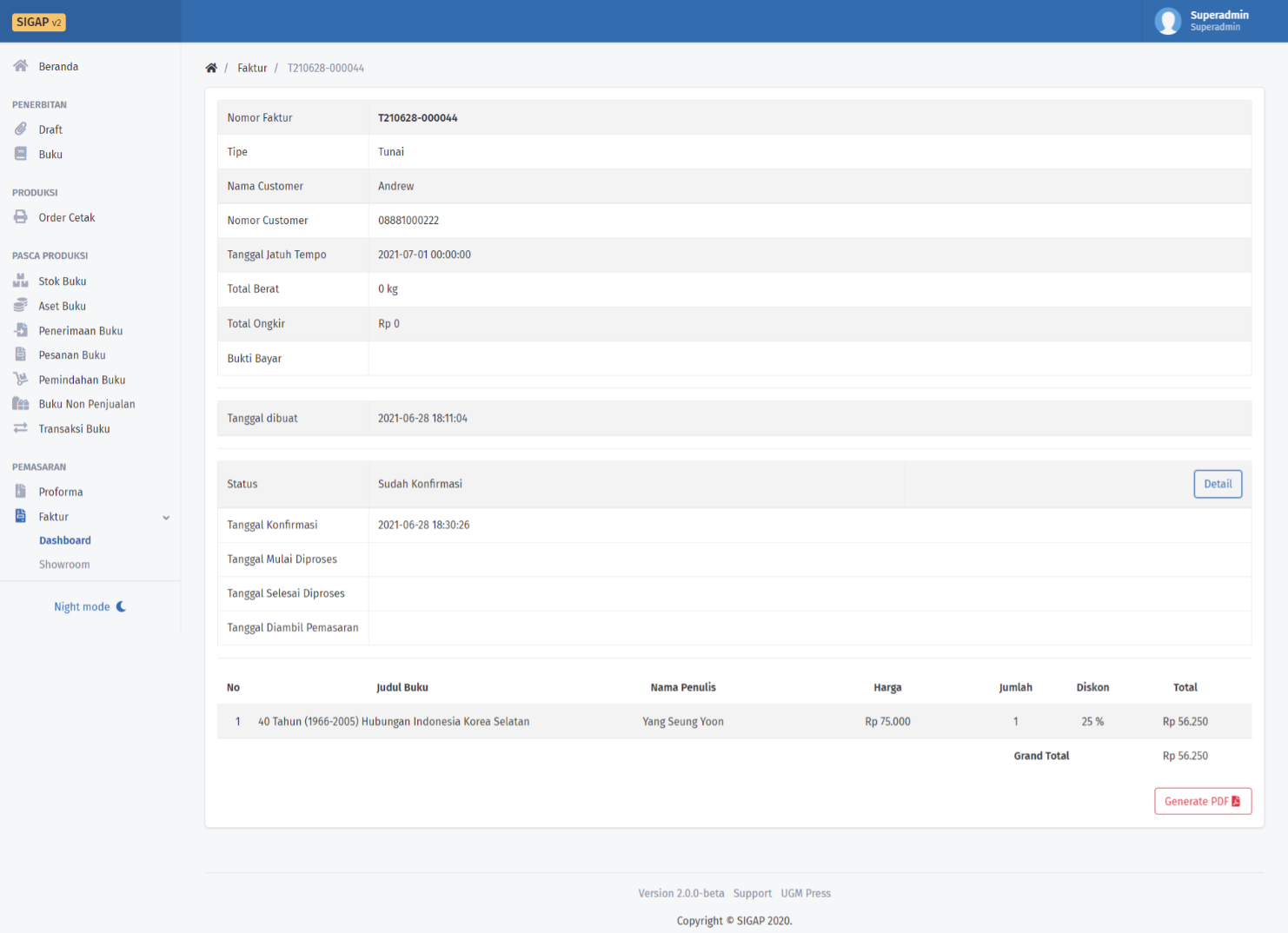
Gambar 10. Keadaan Ketika akan Menyetujui atau Menolak Faktur



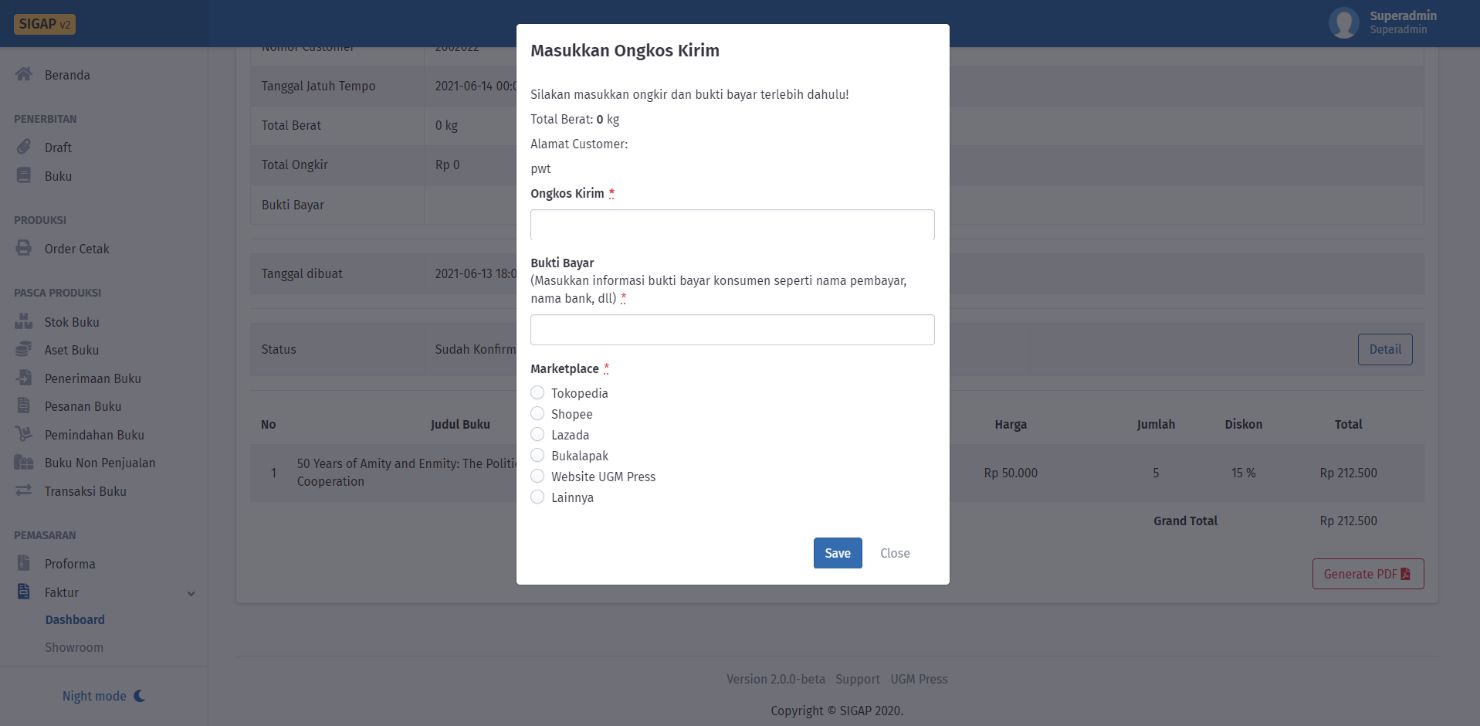
Gambar 11. Halaman *Edit* Faktur



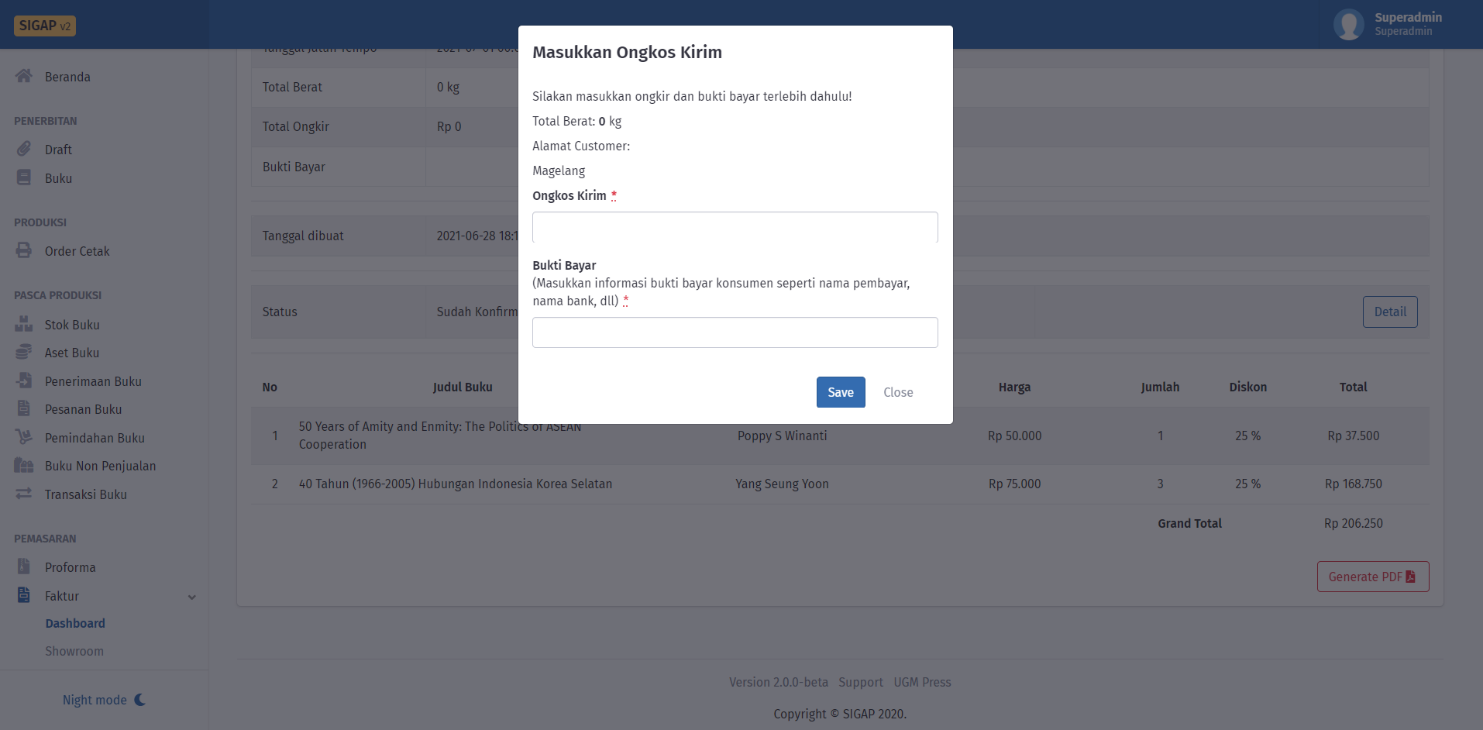
Gambar 12. Halaman Detail Suatu Faktur Ketika Status “Belum Konfirmasi”



Gambar 12. Halaman Detail Suatu Faktur Ketika Status “Sudah Konfirmasi”



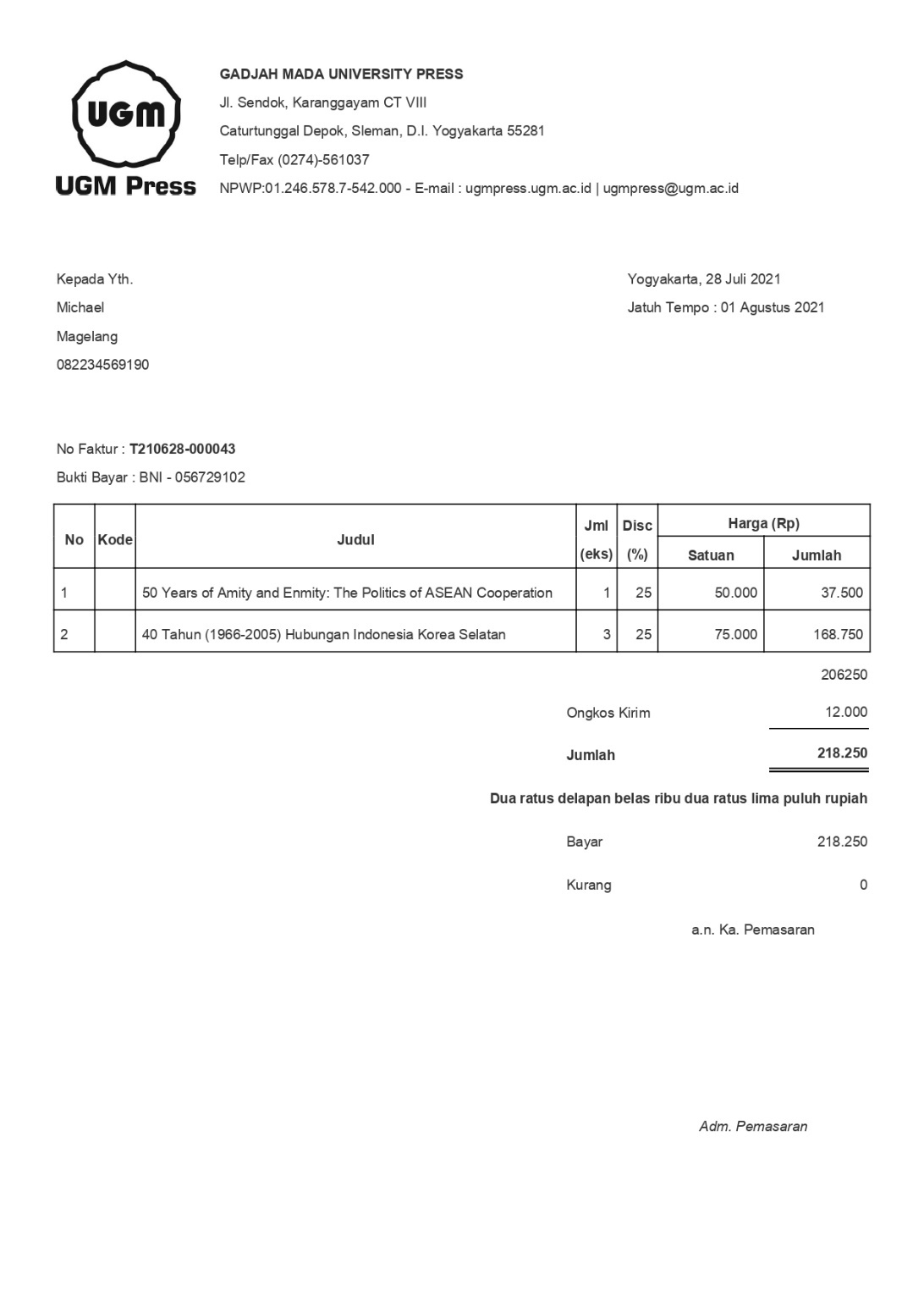
Gambar 13. Modal untuk Memasukkan Ongkos Kirim dan Bukti Bayar Ketika Faktur Jenis *Online*



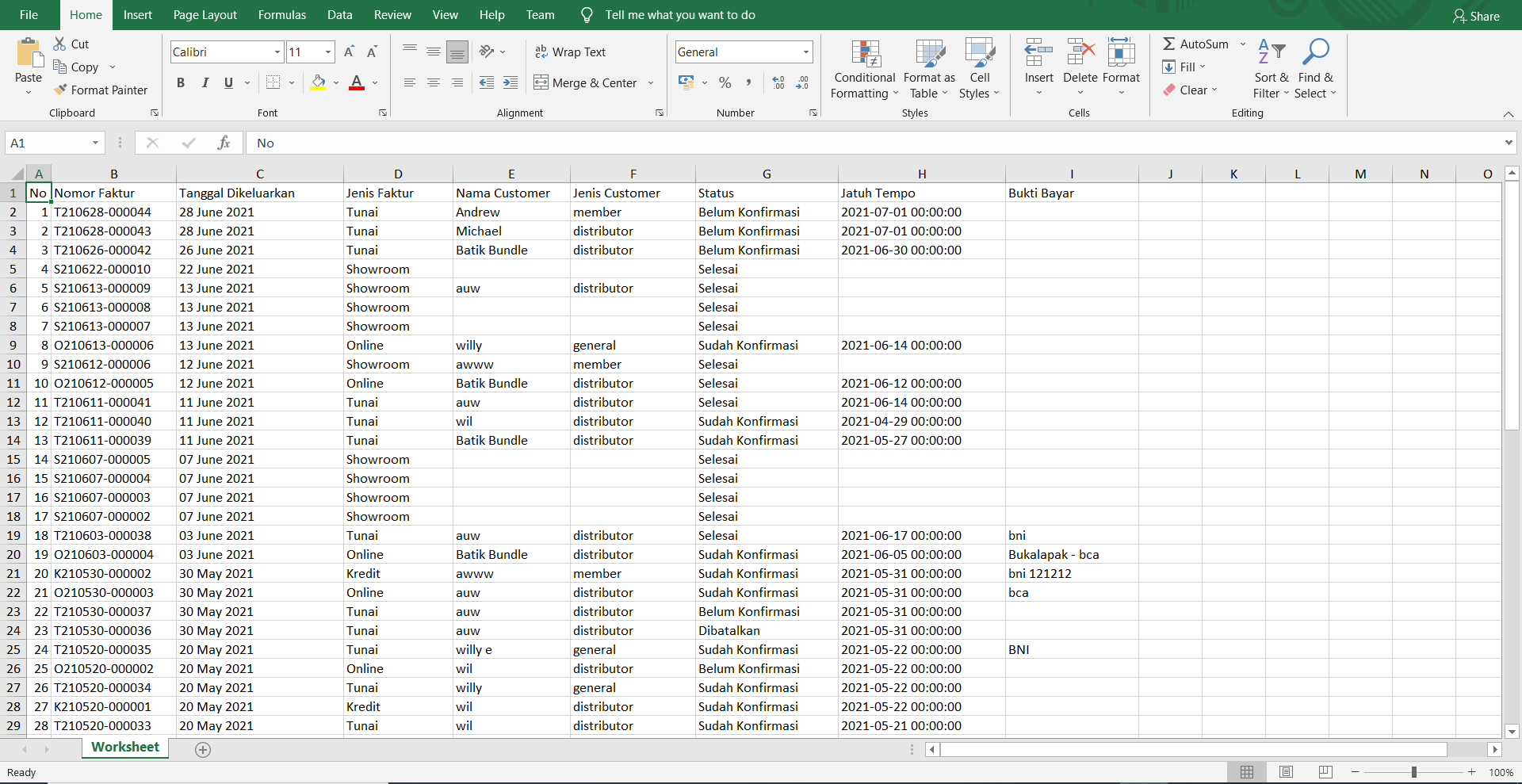
Gambar 14. Modal untuk Memasukkan Ongkos Kirim dan Bukti Bayar Ketika Faktur Jenis Tunai dan Kredit



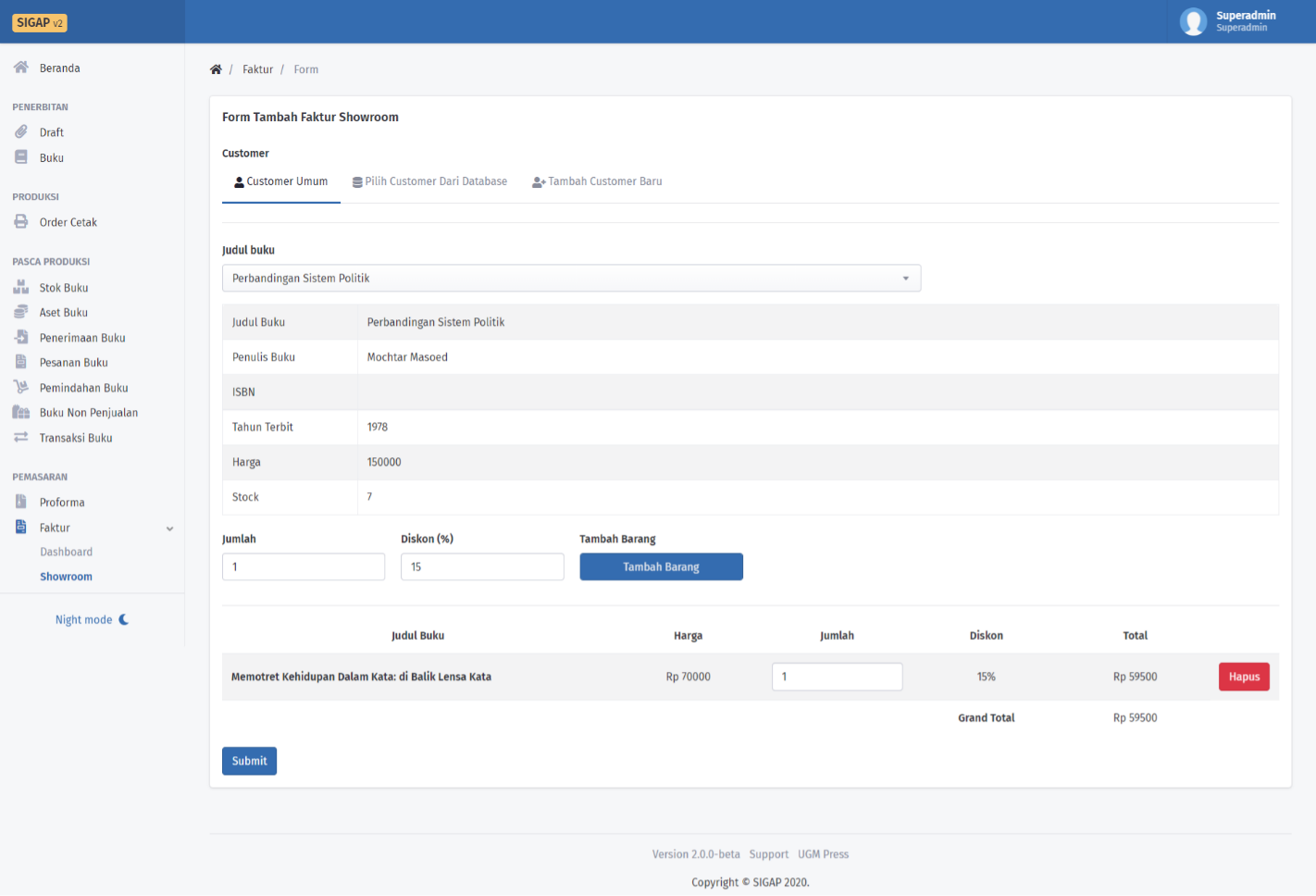
Gambar 15. Modal untuk Menyelesaikan Transaksi Faktur



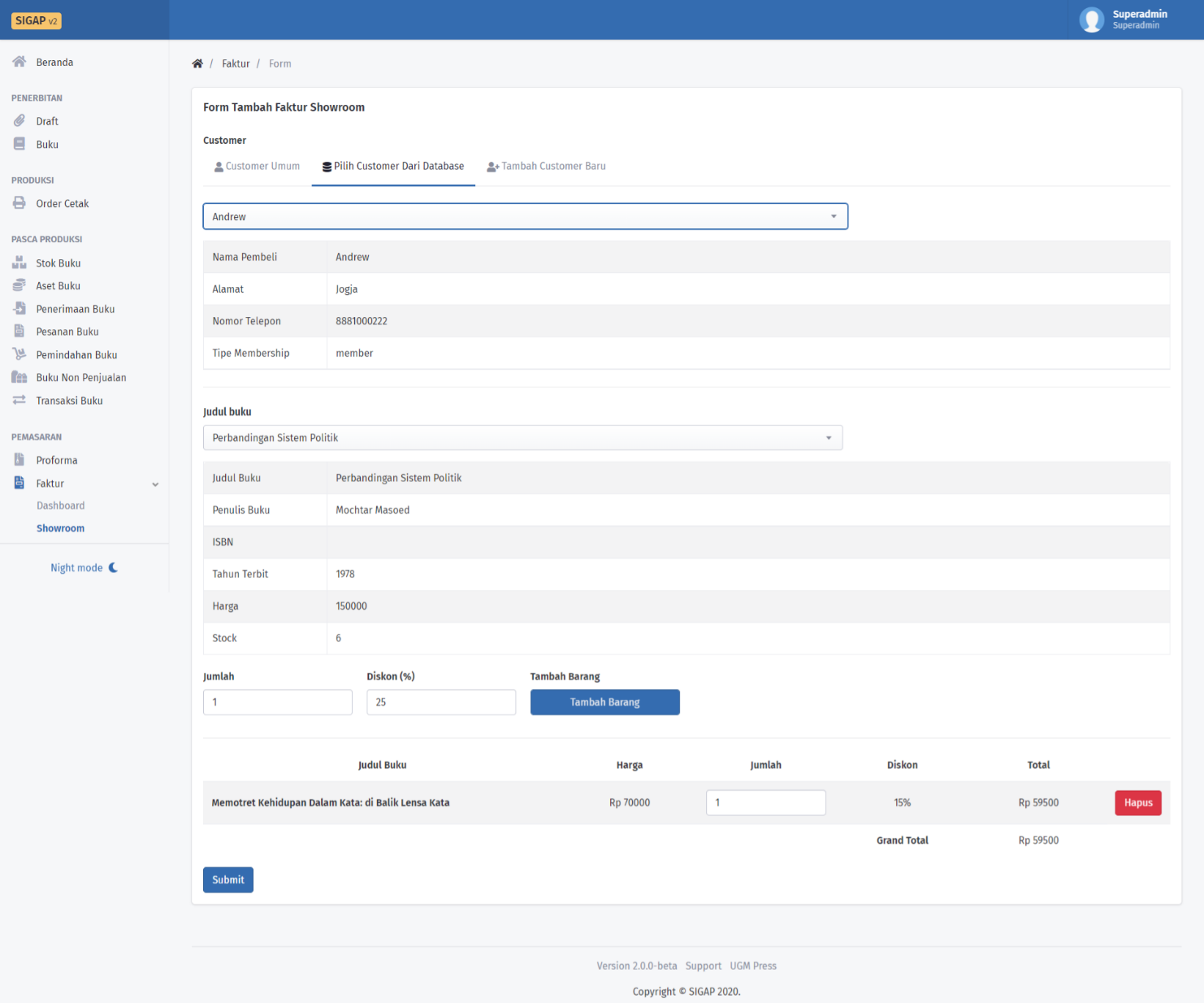
Gambar 16. Tampilan PDF Faktur



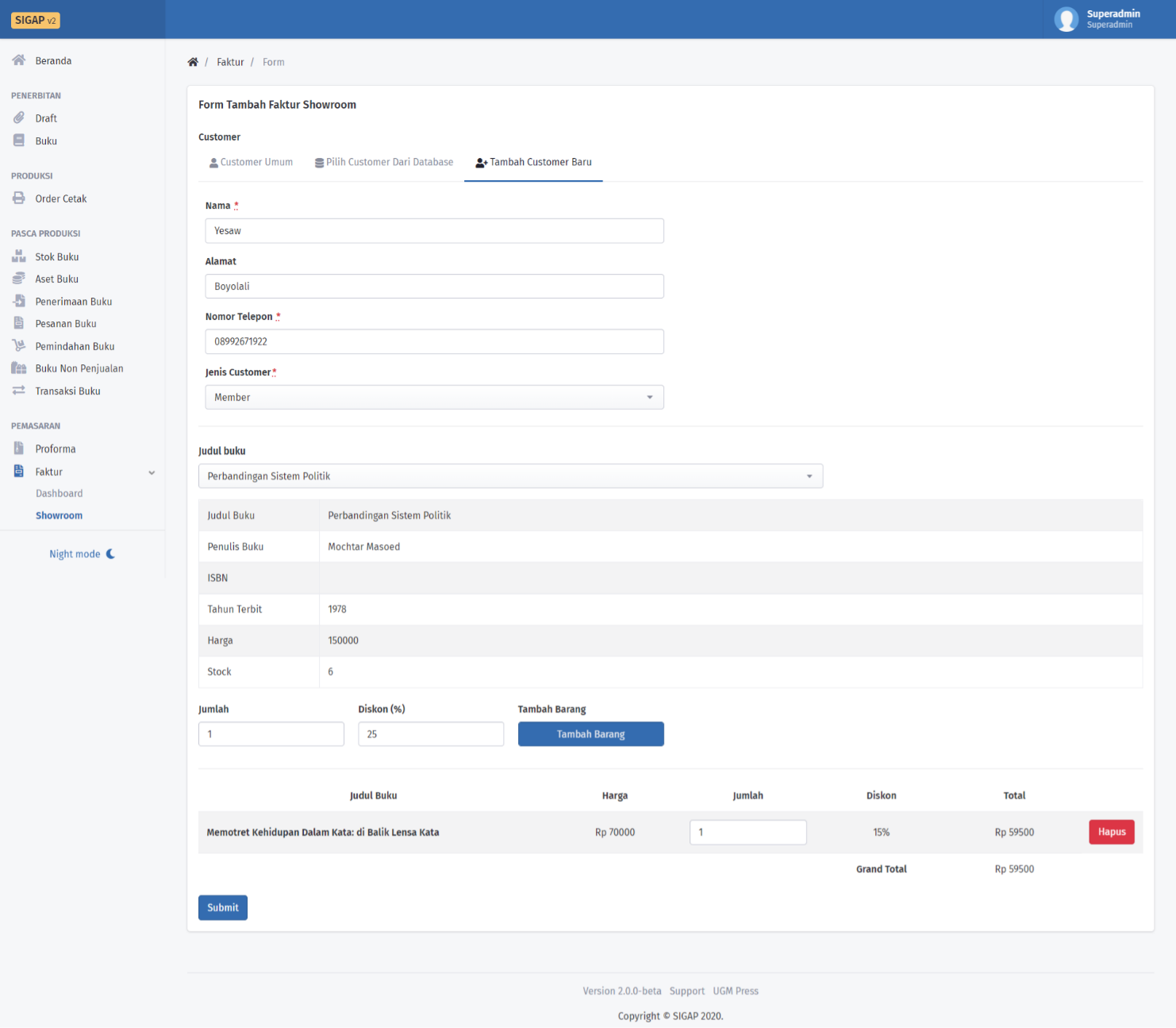
Gambar 17. Tampilan Excel Tabel Faktur



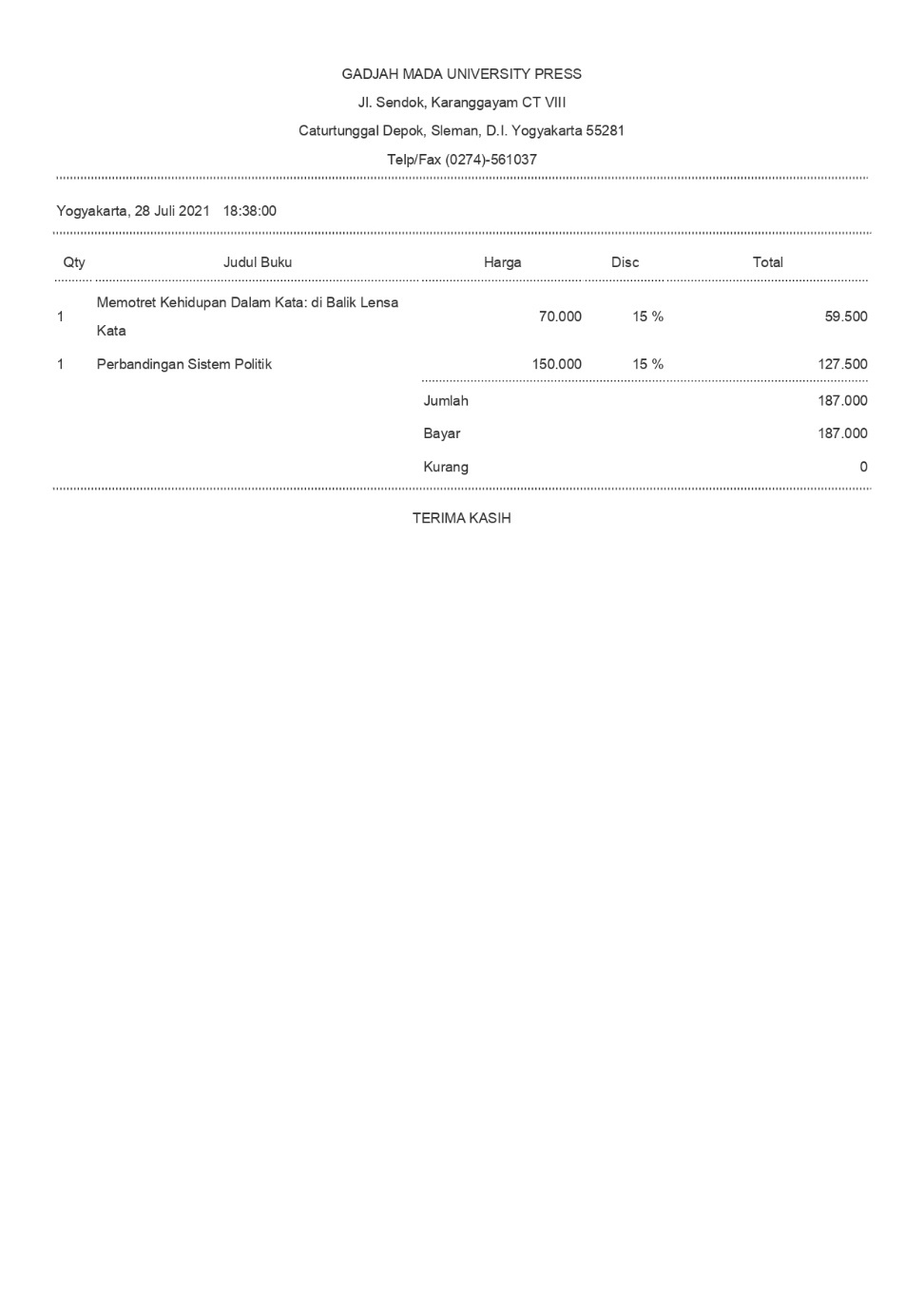
Gambar 18. Halaman Tambah Faktur *Showroom* Ketika Memilih *Customer* Umum



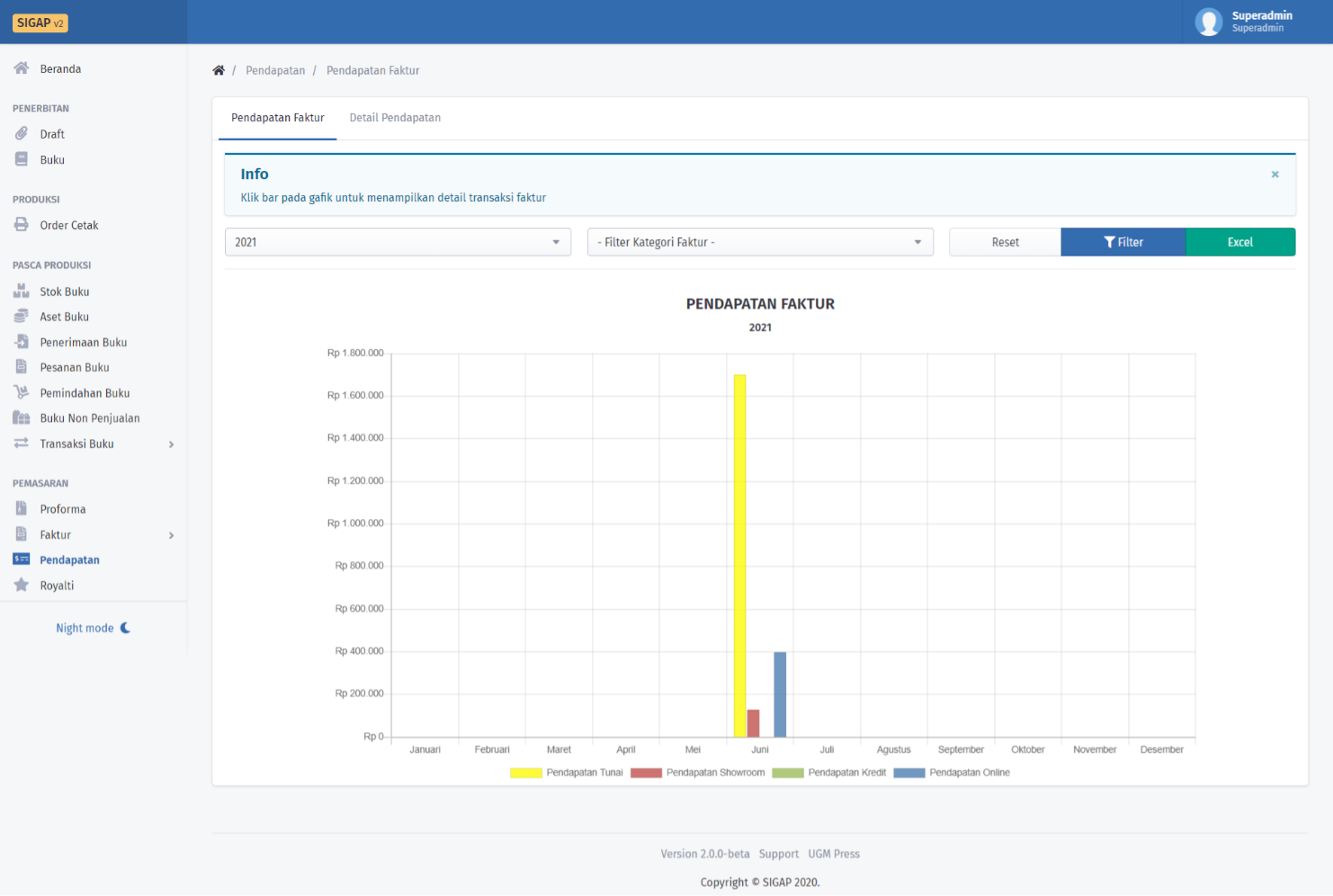
Gambar 19. Halaman Tambah Faktur *Showroom* Ketika Memilih *Customer* dari *Database*



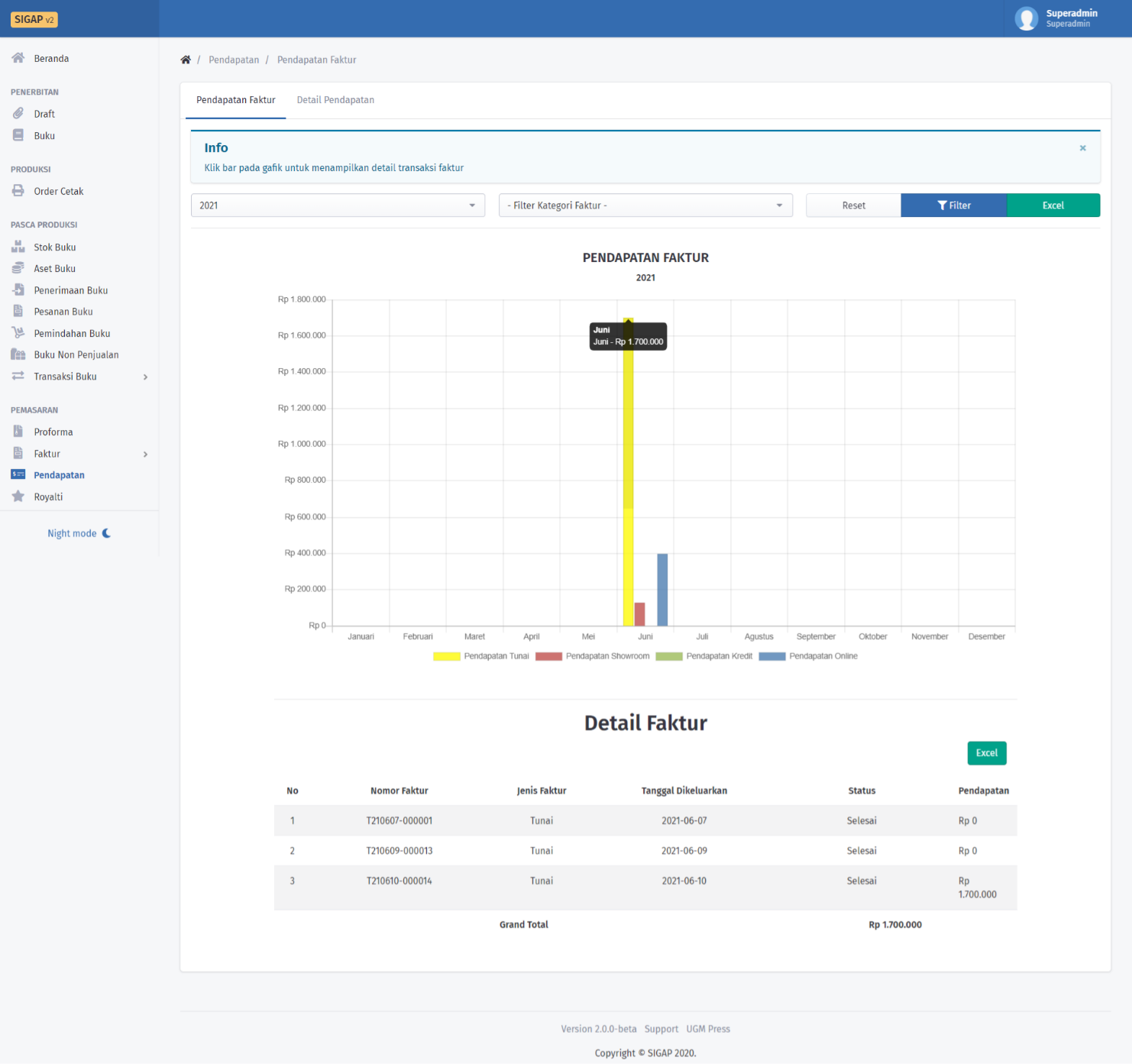
Gambar 20. Halaman Tambah Faktur *Showroom* Ketika Memilih dan Memasukkan *Customer* Baru



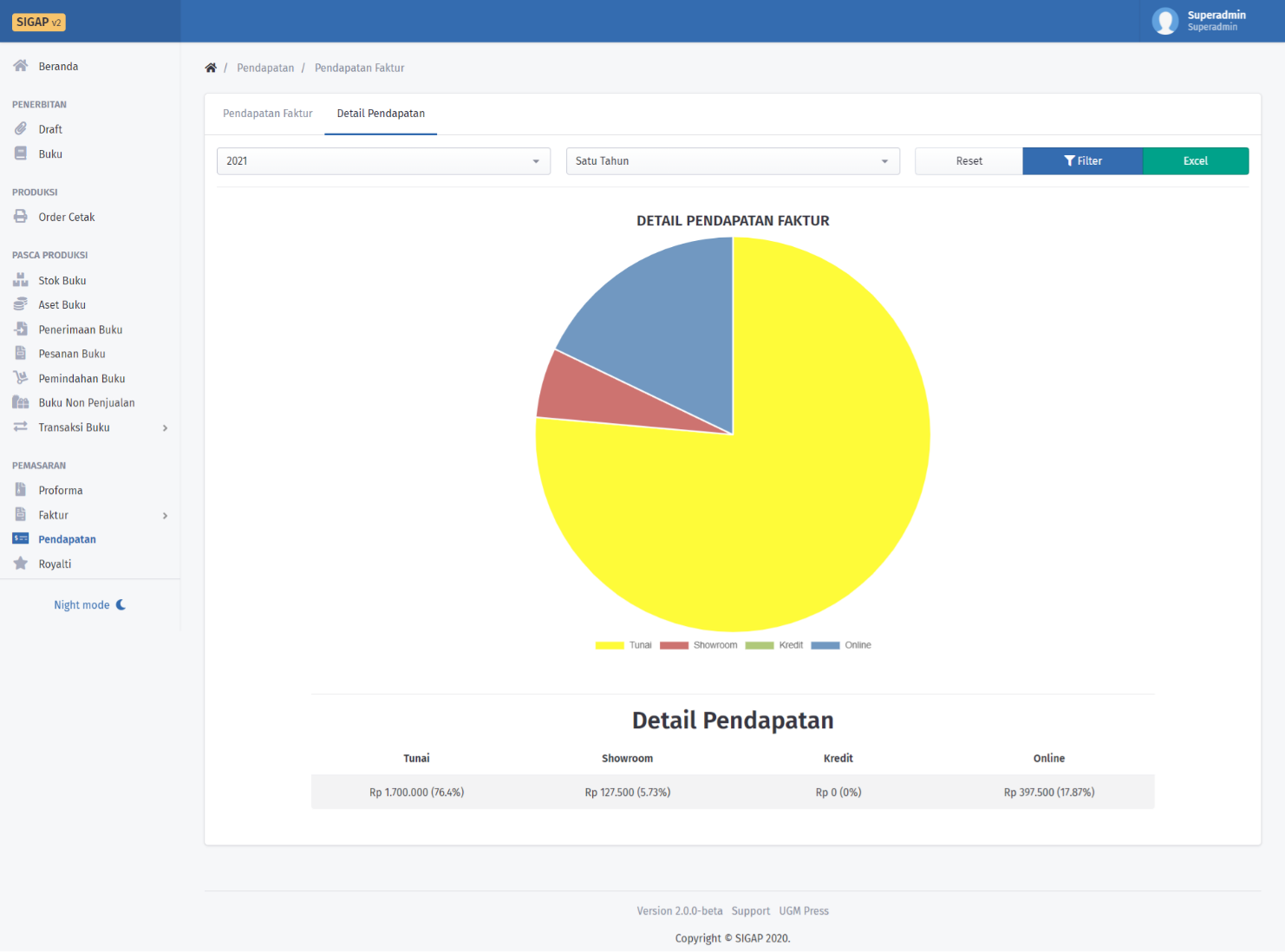
Gambar 21. Tampilan *Receipt Showroom*



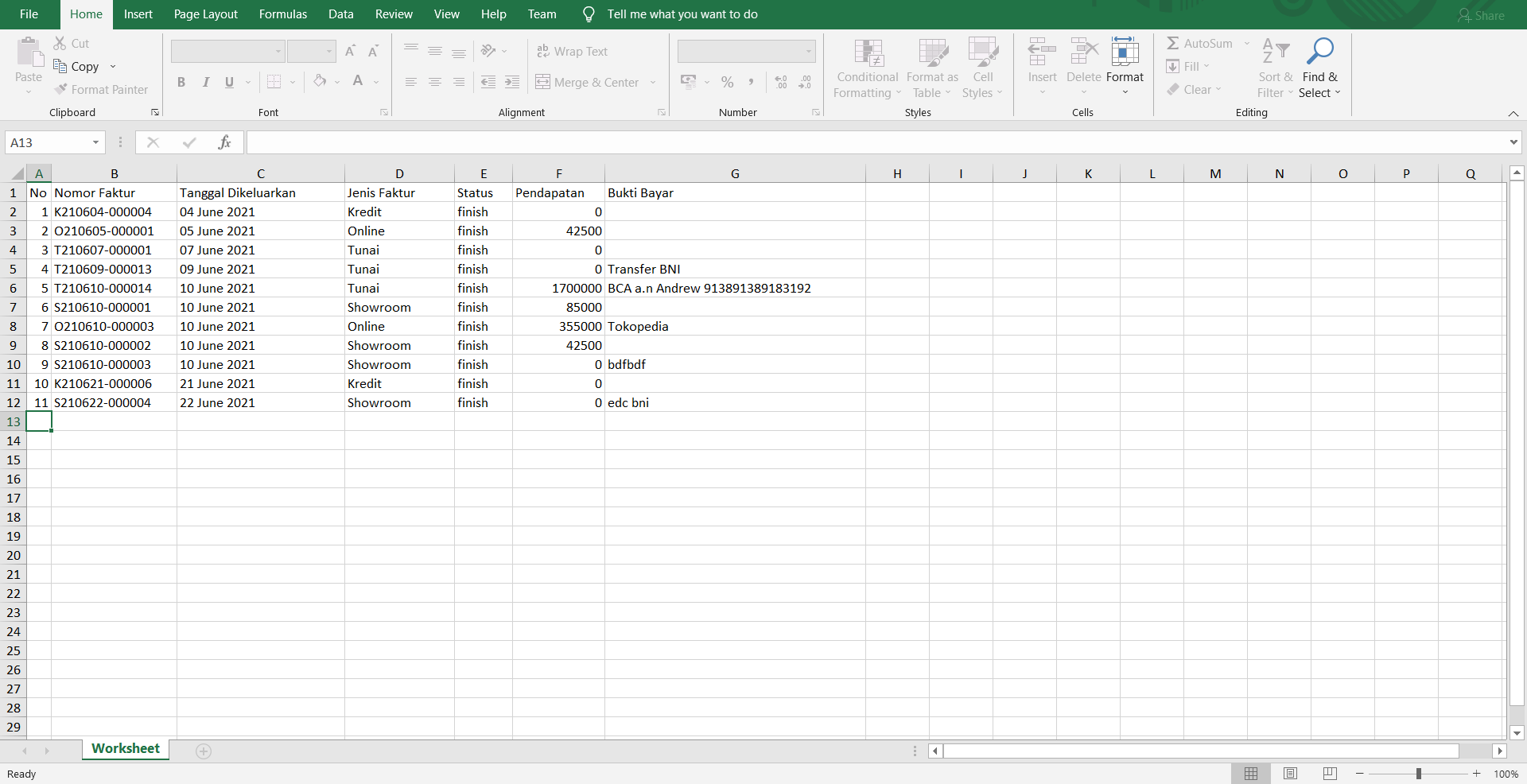
Gambar 22. Halaman Pendapatan Faktur



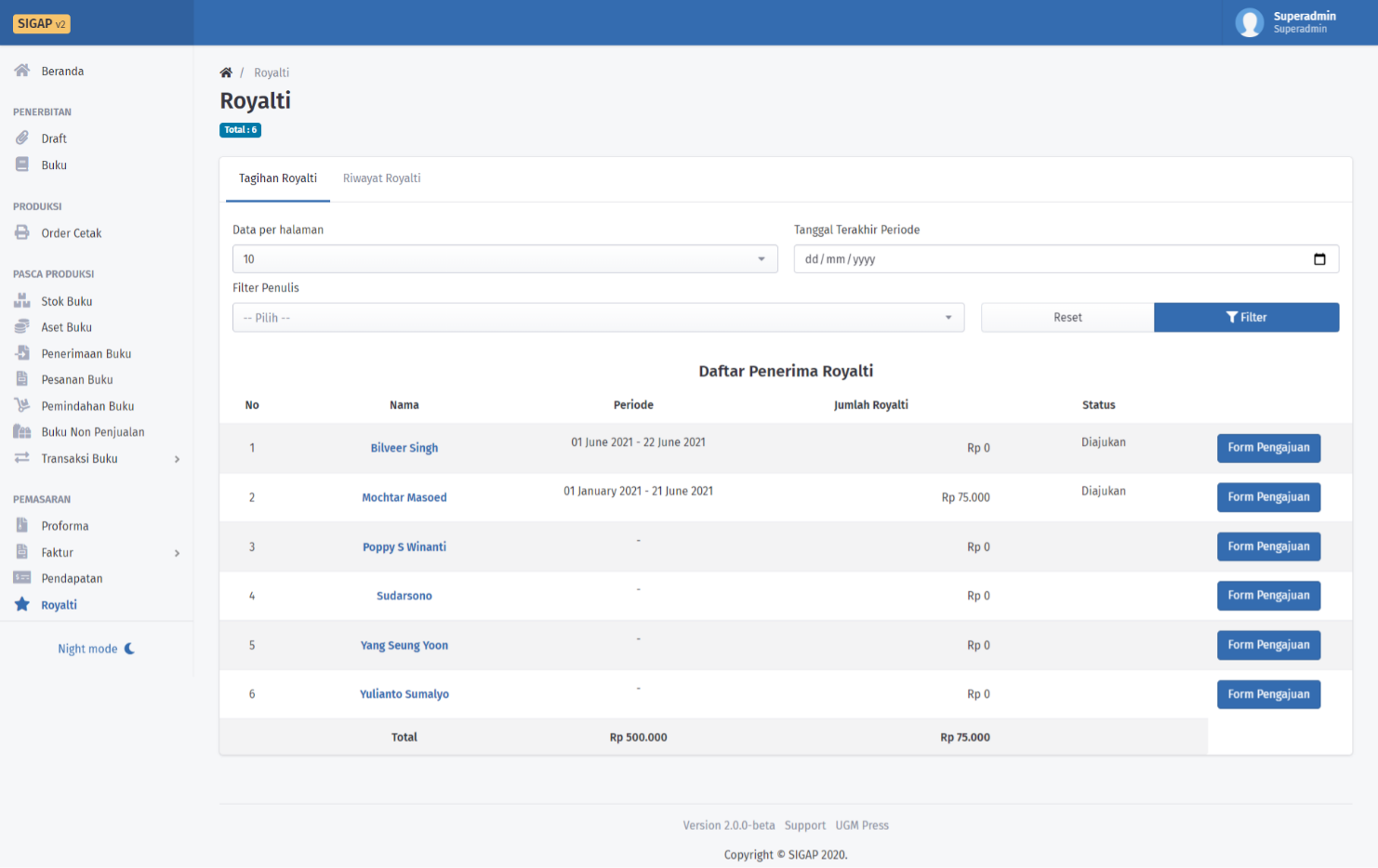
Gambar 23. Tampilan Tabel Detail Pendapatan suatu Jenis Faktur



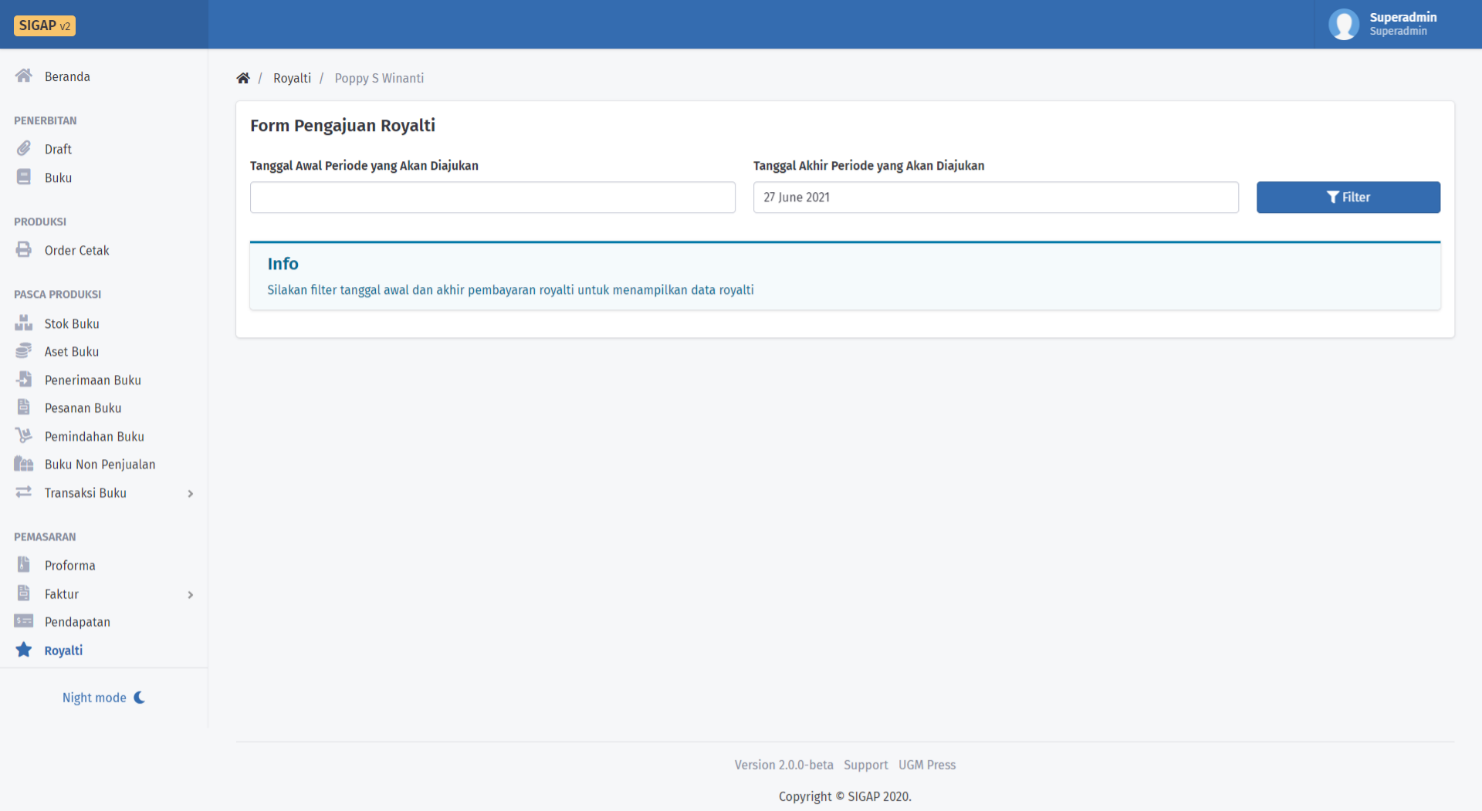
Gambar 24. Halaman Detail Pendapatan



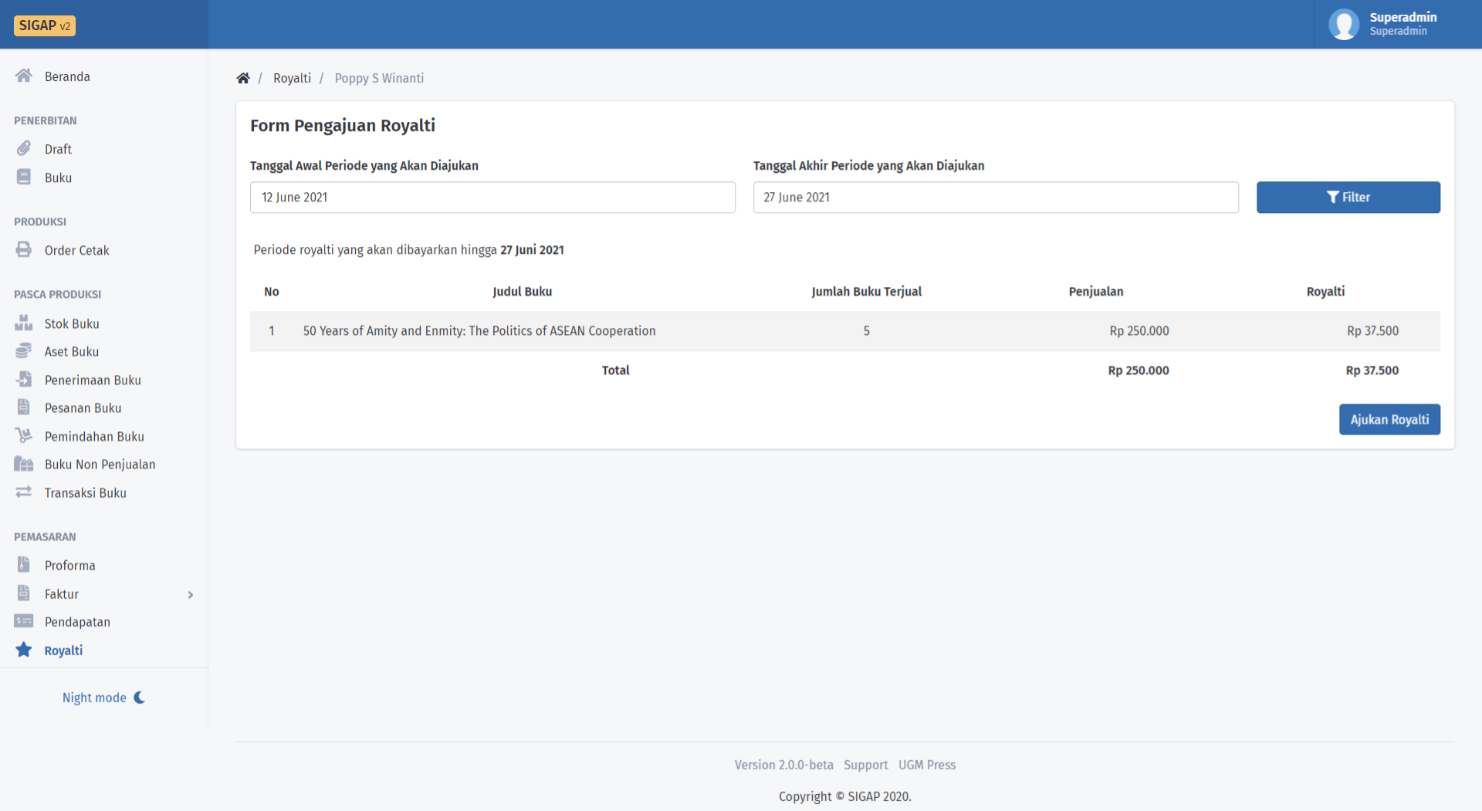
Gambar 25. Tampilan Excel Laporan Pendapatan Faktur

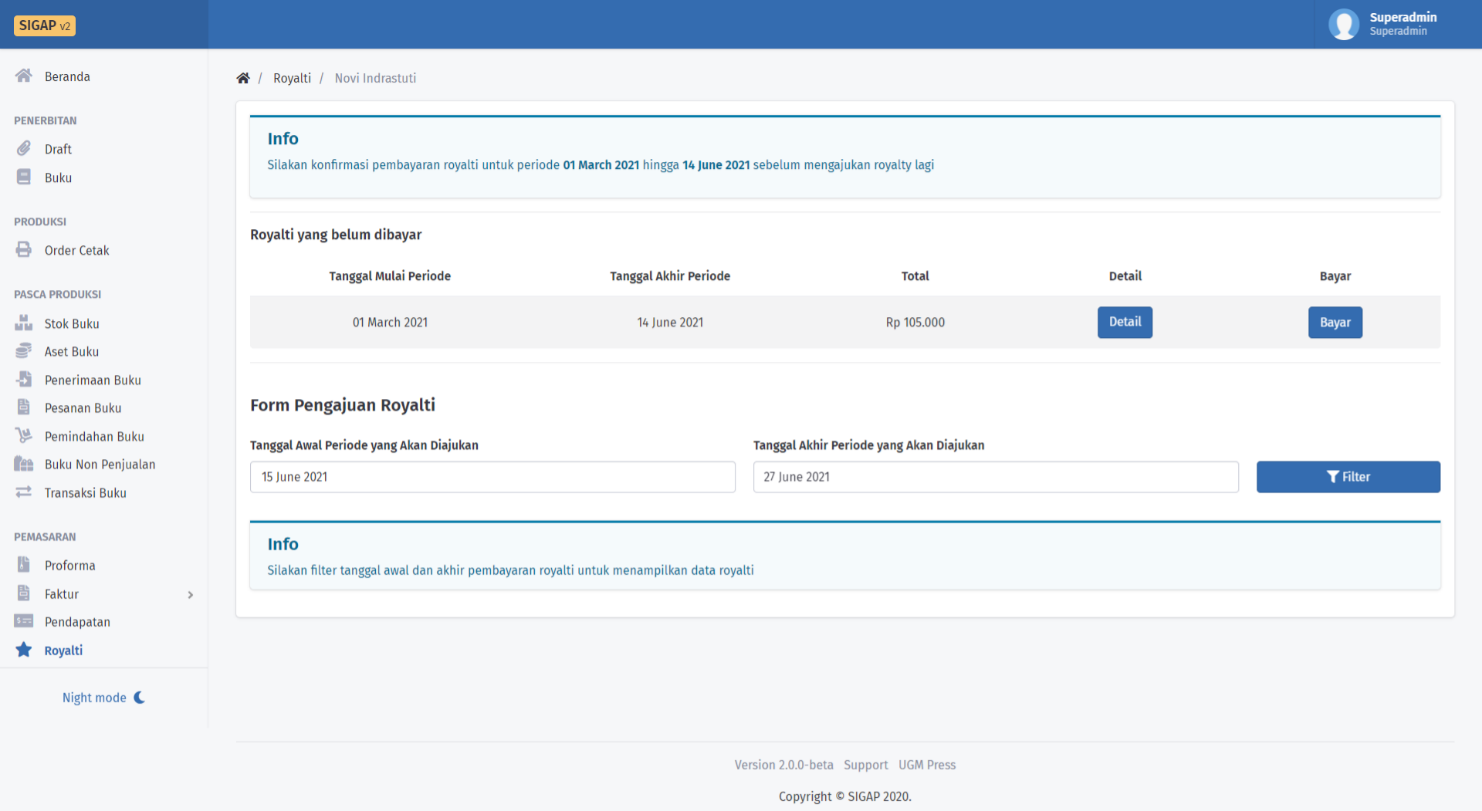


Gambar 26. Halaman Royalti *Tab* “Tagihan Royalti”

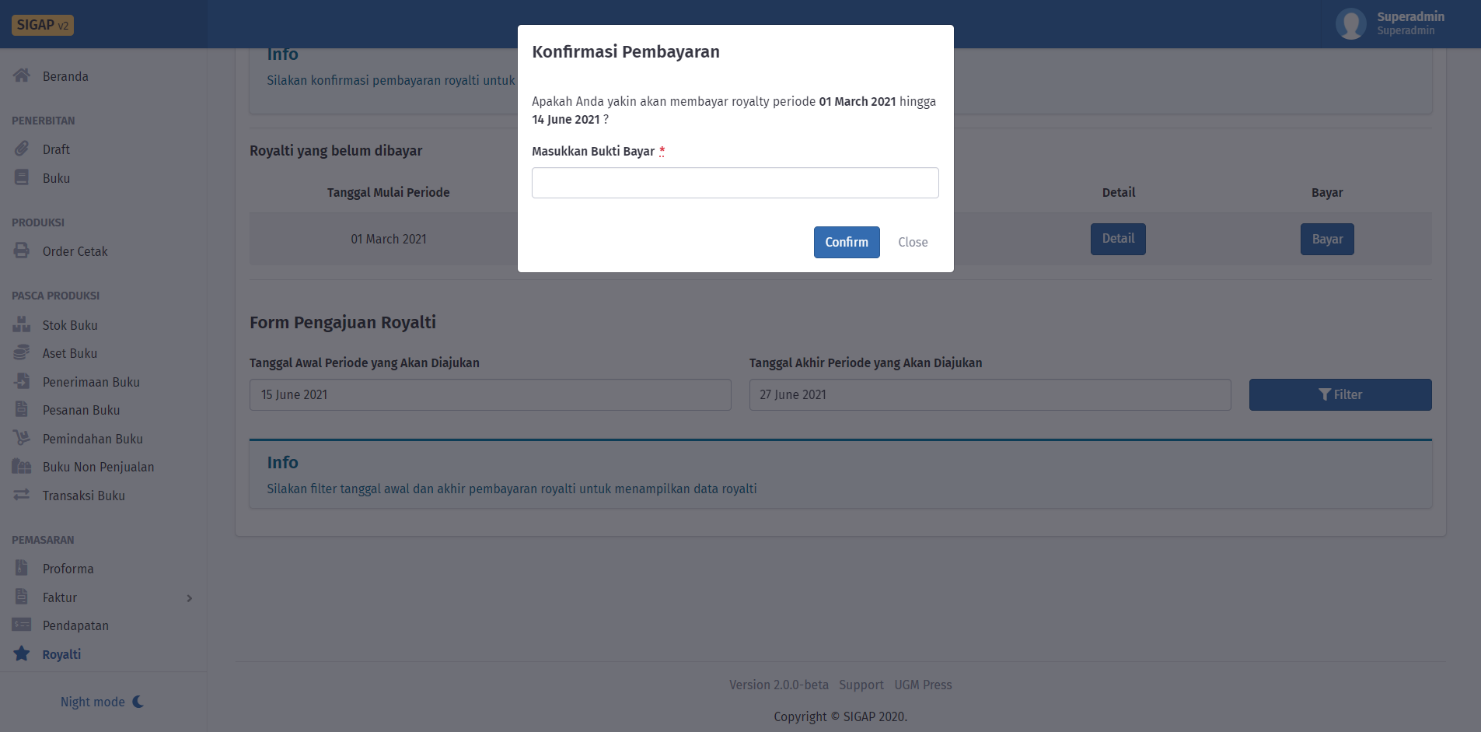


Gambar 27. Halaman Pengajuan Royalti akan Mengajukan Pembayaran Royalti

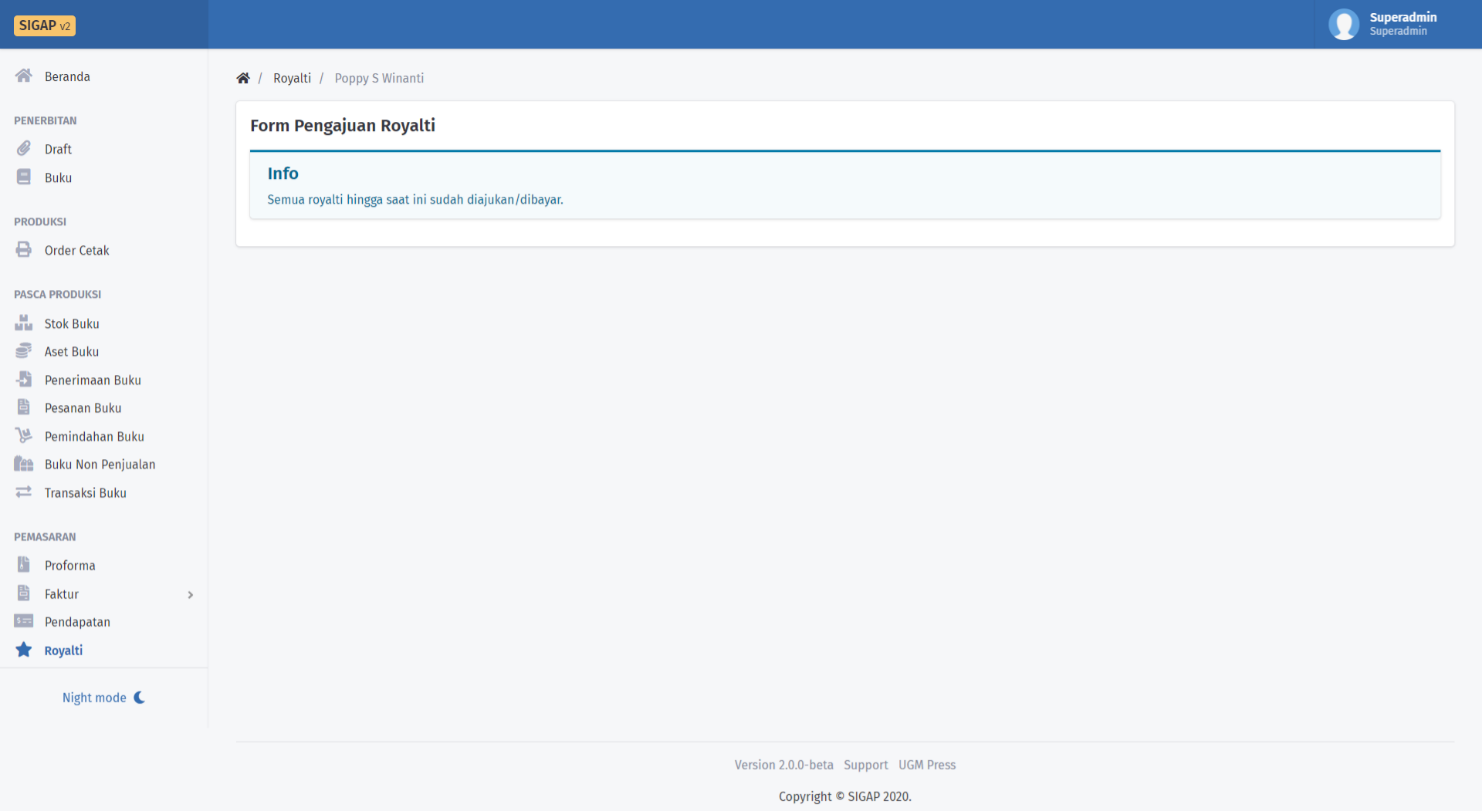
Gambar 28. Halaman Pengajuan Royalti Ketika Sudah Memilih Periode Royalti



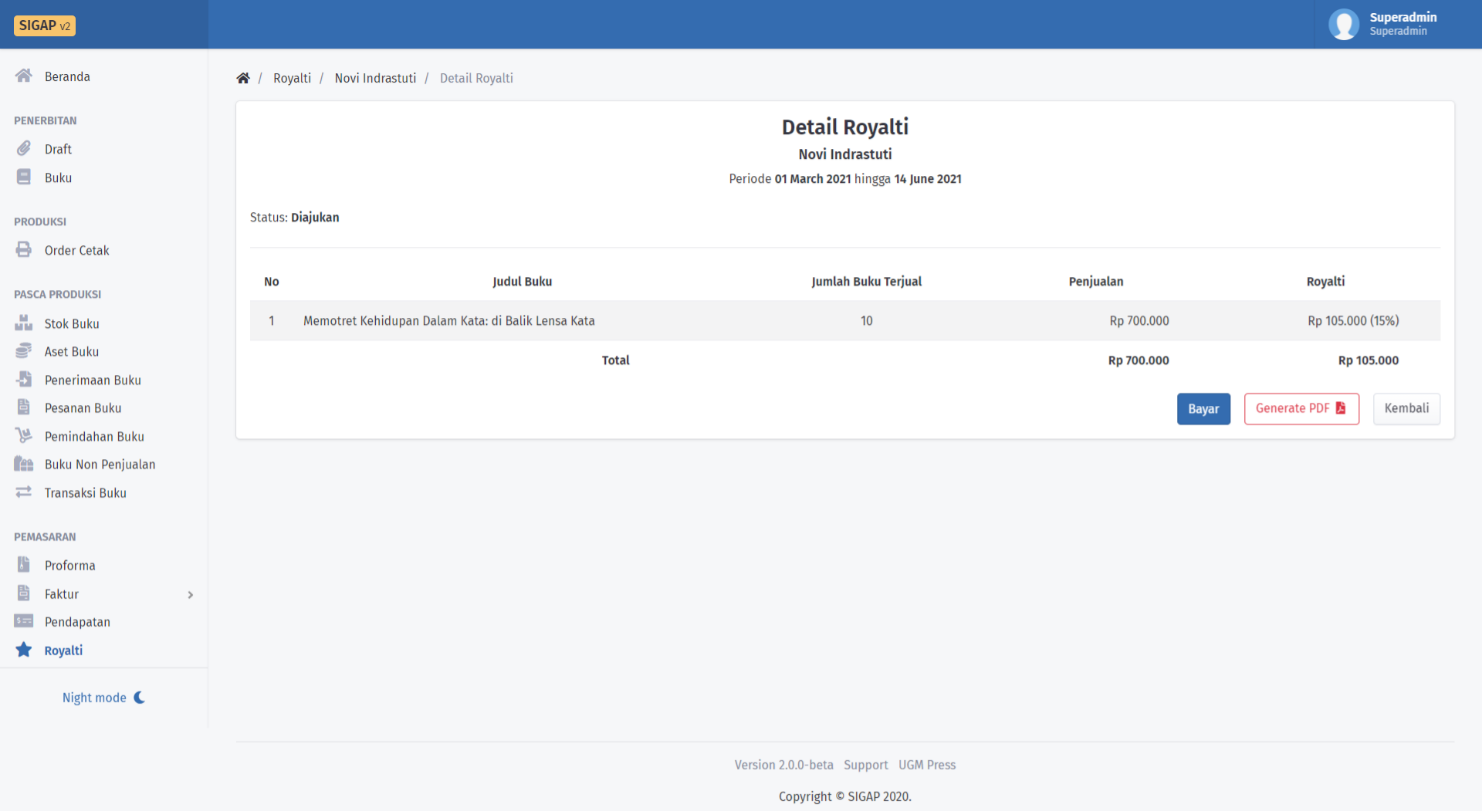
Gambar 29. Halaman Pembayaran Royalti



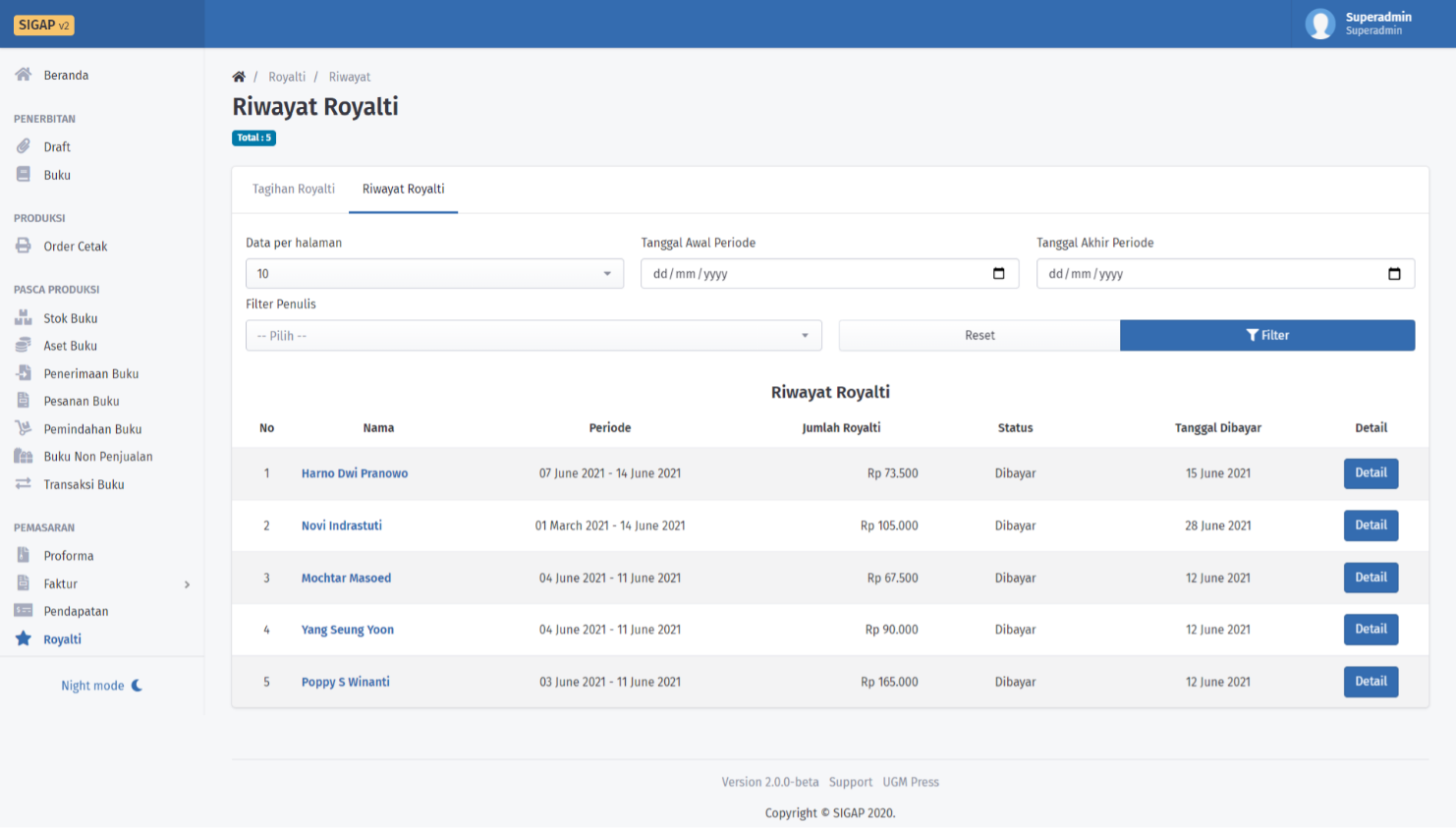
Gambar 30. Modal Konfirmasi Bukti Bayar Pembayaran Royalti



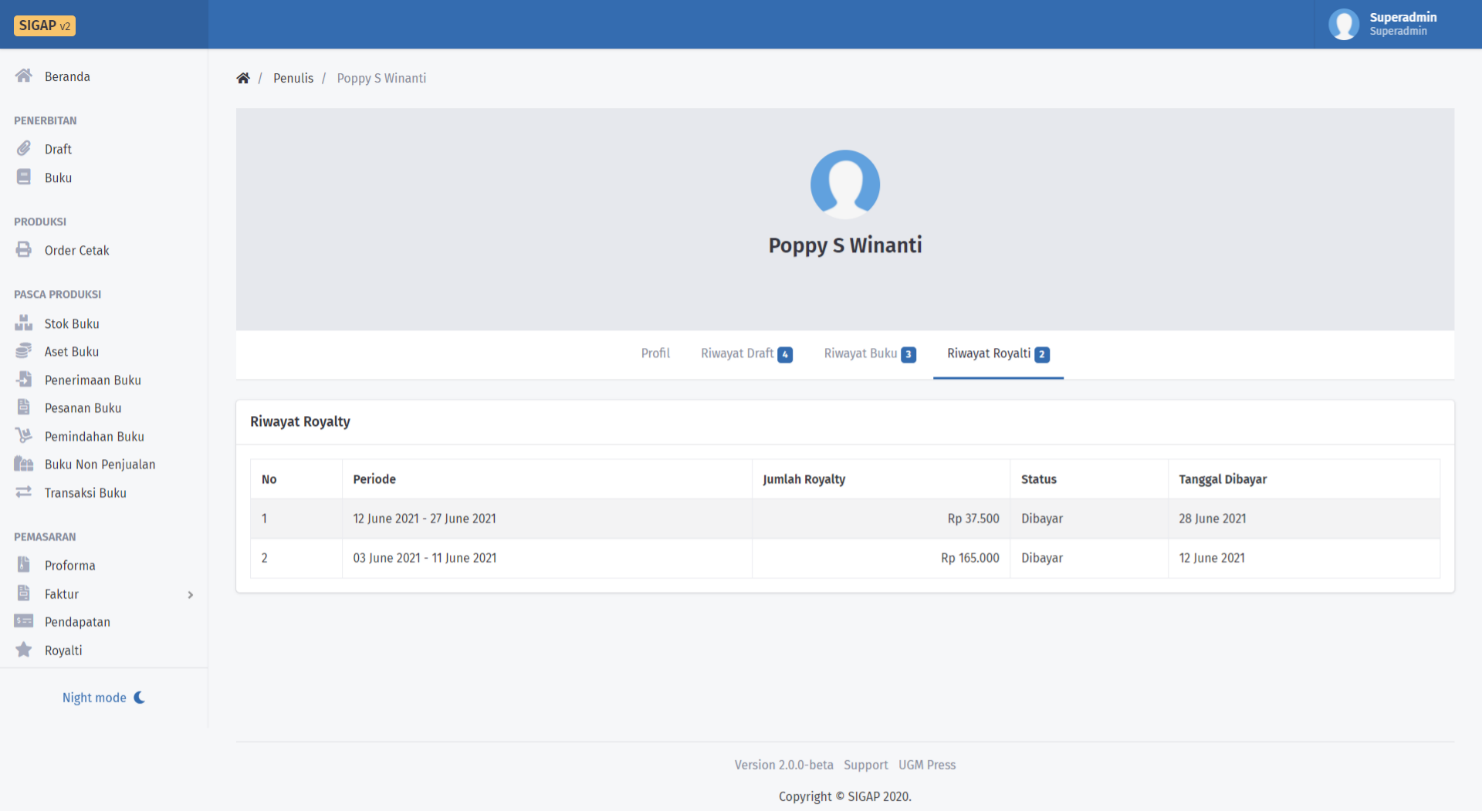
Gambar 31. Halaman Pengajuan Royalti Ketika Sudah Tidak Ada Royalti yang Dapat Diajukan / Dibayar



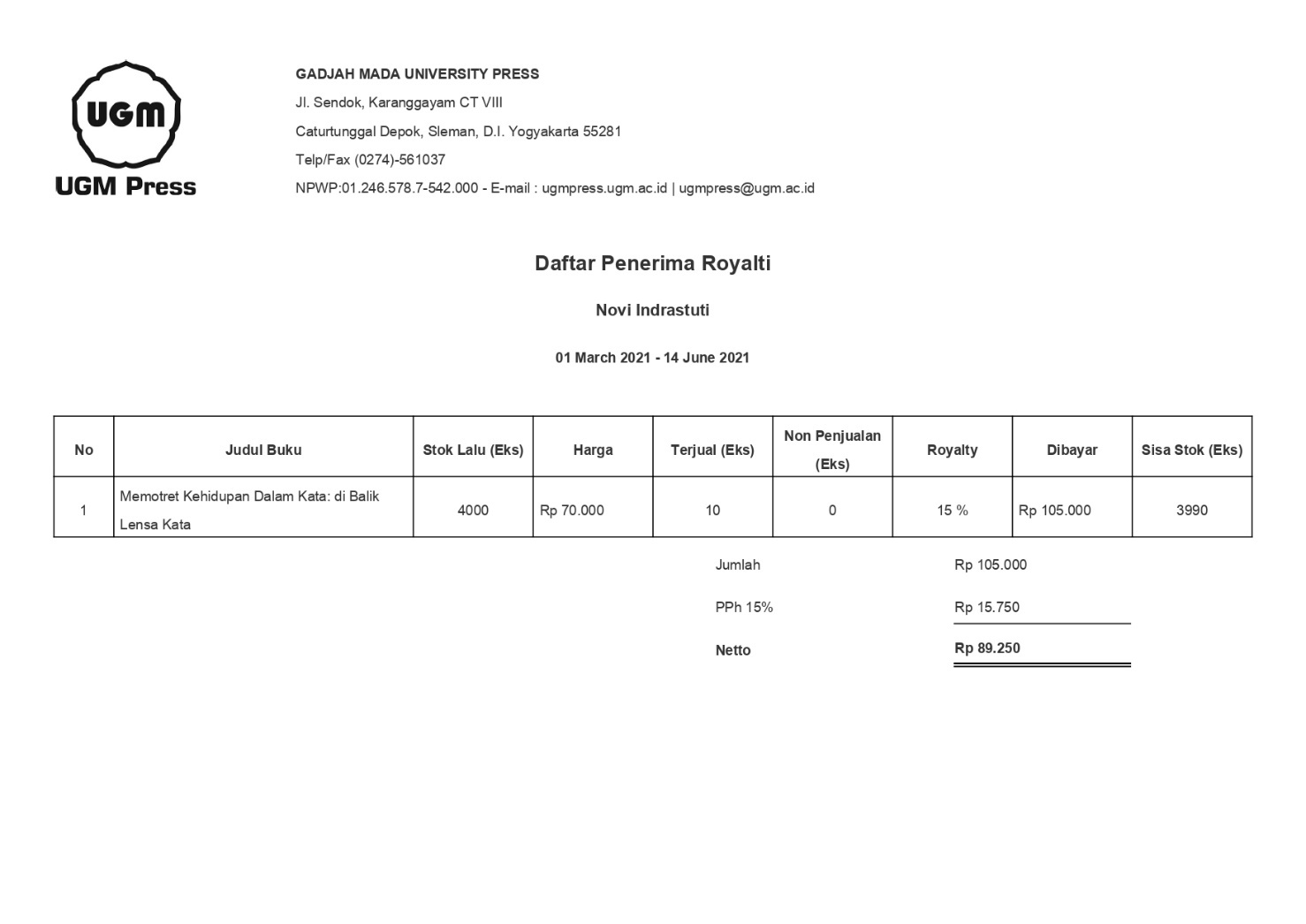
Gambar 32. Halaman Detail Royalti Penulis



Gambar 33. Halaman Royalti *Tab* “Riwayat Royalti”



Gambar 34. Halaman Penulis *Tab* Riwayat Royalti



Gambar 35. Tampilan PDF Laporan Penerimaan Royalti Penulis