**INTEGRASI DAN MIGRASI SISTEM (A)  
LAPORAN TUGAS BESAR SISTEM REMINDER (BAB I- II)**



**Oleh:**

**Group I Kelompok I**

1. **I Komang Wahyu Hadi Permana (1905551010)**
2. **Prana Pandu Pratama (1905551015)**
3. **Putu Gede Sanchia Jodie Mantra (1905551028)**
4. **I Putu Jaya Antara (1905551029)**
5. **Vidya Chandradev (1905551067)**
6. **Bagas Gustama (1905551069)**
7. **Dewa Ngakan Made Bagus Krishna K. (1905551079)**
8. **I Nyoman Jyotisa (1905551092)**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2020**

DAFTAR ISI

[DAFTAR ISI i](#_Toc83272658)

[BAB I ENGINE NOTIFIKASI 1](#_Toc83272659)

[1.1 Email Service 1](#_Toc83272660)

[1.1.1 Mailchimp 1](#_Toc83272661)

[1.1.2 AWS Simple Email Service 2](#_Toc83272662)

[1.1.3 Alibabacloud Direct-mail 2](#_Toc83272663)

[1.2 Konfigurasi Engine Notifikasi 3](#_Toc83272664)

[1.2.1 Konfigurasi Engine Email Service 3](#_Toc83272665)

[1.2.2 Pengunduhan SDK 23](#_Toc83272666)

[1.2.3 Pengkodean 24](#_Toc83272667)

[1.2.4 Pembuatan GUI dari Engine Email Service 29](#_Toc83272668)

[1.2.5 Instalasi AAPANEL 31](#_Toc83272669)

[1.2.6 Mengganti Url Default 35](#_Toc83272670)

[1.2.7 Memasang SSl 36](#_Toc83272671)

[1.2.8 Hosting 38](#_Toc83272672)

[1.2.9 Implementasi Cronjob Hosting 41](#_Toc83272673)

[1.3 Pengujian Engine 44](#_Toc83272674)

[1.3.1 Pengujian Pada Service Mailchimp 44](#_Toc83272675)

[1.3.2 Pengujian Pada Service AWS 47](#_Toc83272676)

[1.3.3 Pengujian Pada Service Alibaba 51](#_Toc83272677)

[BAB II SISTEM DAN WEBVIEW PHPRUNNER 56](#_Toc83272678)

[2.1 Tahapan Pembuatan Sistem Pada PHPRunner 56](#_Toc83272679)

[2.1.1 Manajemen Akun Personal 56](#_Toc83272680)

[2.1.2 Manajemen Agenda Grup 57](#_Toc83272681)

[2.1.3 Manajemen Agenda Personal 66](#_Toc83272682)

[*2.1.4* Manajemen Grup 71](#_Toc83272683)

[*2.1.5* Manajemen Transaksi 77](#_Toc83272684)

[2.2 Panduan Pengoperasian Sistem 87](#_Toc83272685)

[2.2.1 Panduan Pengoperasian Admin Puncak 88](#_Toc83272686)

[2.2.2 Panduan Pengoperasian Admin Group 89](#_Toc83272687)

[2.2.3 Panduan Pengoperasian User 115](#_Toc83272688)

# ENGINE NOTIFIKASI

## Email Service

*Email Service* merupakan *tools* atau *engine* yang digunakan untuk mengirimkan *email* secara otomatis. Alat ini dapat digunakan untuk mengirimkan notifikasi kepada pelanggan secara otomatis tanpa harus mengetikan dan mengirimkan satu per satu yang tentunya akan menguras waktu dan tenaga. Di dalam penelitian ini, kami menggunakan tiga *email service*, yaitu Mailchimp, AWS Simple Email Service, dan Alibaba Direct-mail. Berikut merupakan penjelasan dari masing-masing *email service* yang digunakan.

### Mailchimp

Mailchimp merupakan salah satu *email service* yang berbasis *subscription* (yang berarti biaya akan ditarik per bulan). Mailchimp juga menyediakan pembelian *credit* tambahan untuk mengirimkan *email*. Berikut merupakan kalkulasi harga yang dikenakan di Mailchimp.

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

Gambar 1.1 Daftar Harga Service Mailchimp

Gambar di atas merupakan gambar dari daftar harga yang dikenakan oleh Mailchimp. Mailchimp menyediakan *free charge* atau tanpa membayar sama sekali. Tetapi, *free charge* memiliki fitur yang terbatas sehingga disarankan untuk memilih di atas *free charge*.

### AWS Simple Email Service

AWS Simple Email Service merupakan engine email sevice yang disediakan oleh AWS (Amazon Web Service). AWS memberikan engine ini secara gratis untuk mengirimkan email sebanyak 62.000 email per bulan. Setelah mencapai batas bulanan, AWS akan menetapkan tarif untuk pengiriman email setelahnya dengan tarif sebagai berikut.

|  |
| --- |
| Graphical user interface, text  Description automatically generated |

Gambar 1.2 Daftar Tarif yang Dikenakan oleh AWS

Gambar di atas merupakan gambar dari tarif biaya yang dikenakan oleh AWS ketika pengiriman email gratis per bulan telah mencapai batas. Jika ditelaah dari tarif tersebut, AWS menggunakan sistem *pay as you go* yang hanya akan menarik biaya kepada pelanggan apabila pelanggan membutuhkan *engine* ini saja.

### Alibabacloud Direct-mail

Alibaba Direct-mail merupakan *engine email service* yang disediakan oleh Alibabacloud. Alibabacloud memberikan *engine* ini secara gratis untuk mengirim total 200 *email* per hari. Setelah mencapai batas harian, Alibabacloud akan menarik biaya untuk pengiriman email selanjutnya. Berikut merupakan tarif yang dikenakan oleh Alibabacloud.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |

Gambar 1.3 Tarif yang Dikenakan oleh Alibabacloud

Gambar di atas merupakan gambar dari tarif yang dikenakan oleh Alibabacloud. Setelah mencapai batas harian, Alibabacloud akan menarik sejumlah $0,29 USD untuk setiap 1000 *email*. Alibabacloud juga menyediakan pilihan berupa *subscription* yang akan ditarik setiap 6 bulan sekali.

## Konfigurasi Engine Notifikasi

*Engine* notifikasi adalah *engine* yang berperan dalam mengirimkan notifikasi. Notifikasi yang dikirimkan adalah dalam bentuk *email*. Proses yang dilalui dalam pembuatan *engine* notifikasi adalah konfigurasi *engine email service*, pengunduhan SDK, pengkodean, pembuatan GUI dari *engine email service*, *hosting*, penempatan *engine* untuk mengirim *email* pada cronjob, dan pengujian. Berikut merupakan pembahasan dari masing-masing proses yang dilalui untuk membentuk *engine* notifikasi.

### Konfigurasi Engine Email Service

Sebelum *engine* dapat digunakan, pelanggan diharuskan untuk melakukan konfigurasi. Konfigurasi ini bermacam-macam, tergantung dari penyedia layanan. Umumnya konfigurasi dilakukan pada *domain* yang akan digunakan untuk mengirim *email* dan *console* dari *engine* yang digunakan. Berikut merupakan konfigurasi yang harus dilakukan di masing-masing penyedia layanan yang digunakan

* 1. Mailchimp

Agar *engine* Mailchimp dapat digunakan, beberapa konfigurasi harus dilakukan. Berikut merupakan konfigurasi yang harus dilakukan beserta penjelasan bagaimana cara melakukan konfigurasi tersebut

* + 1. Pembuatan API Key

API *key* merupakan kunci agar API dapat digunakan dengan baik dan lancar. Pertama, masuk ke dalam menu *profile* yang ada di dalam menu *dashboard* Mailchimp. Cara untuk masuk ke dalam menu *profile* dapat dilihat di gambar berikut.

|  |
| --- |
| Graphical user interface, text  Description automatically generated |

Gambar 1.4 Dashboard Mailchimp

Gambar di atas meupakan gambar dari *dashboard* Mailchimp. Untuk menuju ke menu *profile*, pengguna dapat menekan tombol foto *user* yang sedang *login* dan ketika menu lainnya sudah ter-*popup*, pengguna dapat menekan tombol *profile*.

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

Gambar 1.5 Tampilan Profile Mailchimp

Gambar di atas merupakan tampilan dari menu *profile* pada *dashboard* Mailchimp. Terdapat beberapa menu yang disediakan dalam menu *profile dashboard* Mailchimp. Setelah menu *profile* ditampilkan, pengguna memilih menu Extra, lalu menekan tombol sub menu API Keys.

|  |
| --- |
| Graphical user interface, text, website  Description automatically generated |

Gambar 1.6 Tampilan Sub Menu API Key

Gambar di atas merupakan tampilan dari sub menu API Keys. Pada menu ini ditampilkan semua API *key* yang telah dibuat. Untuk membuat API *key*, pengguna dapat menekan tombol “Create A Mandrill API Key”. Nanti, pengguna akan dilempar ke halaman Mandrillapp.

|  |
| --- |
| A picture containing text, screenshot, monitor  Description automatically generated |

Gambar 1.7 Tampilan Dashboard Mandrillapp

Gambar di atas merupakan tampilan dari halaman Mandrillapp. Di *dashboard* inilah pengguna dapat menambahkan API *key* yang akan digunakan untuk mengirimkan *email* melalui Mailchimp. Untuk menambahkan API *key*, pengguna dapat menekan tombol biru yang bertuliskan “New API Key”.

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

Gambar 1.8 Form Pembuatan API Key

Gambar di atas merupakan *form* ketika tombol “New API Key” ditekan. Pengguna dapat mengisi deskripsi dari API *key* yang diinginkan, IP *address* yang dapat menggunakan API *key* yang telah dibuat, dan *request* yang diinginkan.

|  |
| --- |
| A picture containing text, screenshot, monitor  Description automatically generated |

Gambar 1.9 Daftar API Key

Gambar di atas merupakan gambar dari daftar API *key* yang telah dibuat. Pengguna dapat menyalin API *key* yang telah ditampilkan di tabel tersebut, mengedit API *key* yang terdaftar, serta mengaktifkan/menonaktifkan API *key* yang terdaftar.

* + 1. Konfigurasi Domain di Mandrill

Setelah mendapatkan API *key*, pengguna harus melakukan konfigurasi *domain* yang digunakan untuk mengirimkan *email*. Pertama, pada halaman Mandrillapp yang sudah disinggung sebelumnya, pengguna menekan menu Domains yang terletak pada bagian atas atau *header*.

|  |
| --- |
|  |

Gambar 1.10 Menu Domain pada Mandrillapp

Gambar di atas merupakan gambar dari menu Domains yang ditekan sebelumnya, Pengguna dapat langsung mengisi alamat *domain* yang digunakan untuk mengirim *email*.

Setelah pengguna mengisi alamat *domain* yang digunakan untuk mengirim *email,* pengguna harus melakukan konfigurasi DKIM dan SPF yang nantinya akan dikonfigurasi di CPanel *domain* tersebut. Untuk konfigurasi DKIM dan SPF dapat dilihat di kode program berikut

|  |
| --- |
| //DKIM, nama txt recordnya mandrill.domainkey.namaDomain  v=DKIM1; k=rsa; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCrLHiExVd55zd/IQ/J/mRwSRMAocV/hMB3jXwaHH36d9NaVynQFYV8NaWi69c1veUtRzGt7yAioXqLj7Z4TeEUoOLgrKsn8YnckGs9i3B3tVFB+Ch/4mPhXWiNfNdynHWBcPcbJ8kjEQ2U8y78dHZj1YeRXXVvWob2OaKynO8/lQIDAQAB;  //jika DKIM memerlukan semicolons, gunakan value ini untuk DKIM  v=DKIM1\; k=rsa\; p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCrLHiExVd55zd/IQ/J/mRwSRMAocV/hMB3jXwaHH36d9NaVynQFYV8NaWi69c1veUtRzGt7yAioXqLj7Z4TeEUoOLgrKsn8YnckGs9i3B3tVFB+Ch/4mPhXWiNfNdynHWBcPcbJ8kjEQ2U8y78dHZj1YeRXXVvWob2OaKynO8/lQIDAQAB\;  //SPF txt record  v=spf1 include:spf.mandrillapp.com ?all |

Kode Program 1.1 Konfigurasi DKIM dan SPF pada Mailchimp

Kode program di atas merupakan konfigurasi yang harus diisi pada konfigurasi DKIM dan SPF. Ketika sudah melakukan konfigurasi SPF dan DKIM, maka *domain* dapat dites pada halaman Mandrillapp.

* + 1. Konfigurasi SPF dan DKIM Pada Domain

Ketika value SPF dan DKIM sudah didapatkan, kiita dapat melakukan konfigurasi dengan cara memasukkan value yang sudah diberikan sebelumnya ke dalam DNS domain kita. Pada kasus kali ini, saya menggunakan domain yang tertaut di Alibaba cloud sehingga konfigurasi SPF dan DKIM akan melalui halaman DNS Alibaba cloud. Berikut adalah penjelasannya.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.11 Halaman Alibabacloud DNS

Gambar di atas merupakan tampilan halaman Alibaba Cloud DNS. Langkah pertama yaitu, masuk ke halaman Alibaba cloud kemudian Login menggunakan akun Alibaba cloud dan masuk ke halaman *console*. Setelah masuk ke halaman *console*, klik menu pada bagian pojok kiri atas dan pilih halaman Alibaba Cloud DNS kemudian akan tampil list domain yang kita meiliki yang tertaut pada Alibaba cloud seperti gambar di atas. Setelah list domain muncul, klik domain yang sudah kita daftarkan pada Mandrill dan akan muncul list konfigurasi dns yang terdapat pada domain tersebut.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.12 List DNS Setting Domain

Gambar di atas merupakan tampilan list Dns setting yang terdapat pada domain yang didaftarkan sebelumnya. Setelah halaman tersebut muncul, cara untuk menambahkan konfigurasi SPF dan DKIM dapat dengan menekan tombol *Add Record* lalu akan muncul sebuah form untuk menambahkan value baru.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.13 Pengisian Value DKIM Mandrill

Gambar di atas merupakan contoh pengisian value DKIM. Dapat dilihat, DKIM memiliki type Text sementara *Host* yang diisikan adalah *host* yang sudah diberikan sebelumnya pada halaman konfigurasi Maindrill begitupula valuenya. Setelah mengisi semua form dan dirasa benar kilik Confirm, maka configurasi DKIM maindril sudah berhasil dilakukan. Setelah DKIM kita akan lanjutkan memasukan konfigurasi SPF.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |

Gambar 1.14 Konfigurasi SPF Mandrill

Gambar di atas merupakan tampilan penambahan value dari konfigurasi SPF Mandrill. Dapat dilihat cara menambahkan SPF hampir sama dengan cara menambahkan DKIM hanya saja value dan *host*nya berbeda. Ketika keduanya sudah di tambahkan maka 2 value yang baru saja kita masukkan tadi akan otomatis terlihat pada list setting dns.

|  |
| --- |
| Graphical user interface, text, application, email  Description automatically generated |

Gambar 1.15 List Value SPF dan DKIM Mandrill

Gambar di atas merupakan tampilan list yang sudah berhasil dimasukkan value SPF dan DKIM dari Mandrill. Setelah list tersebut muncul, user dapat melakukan pengecekan Kembali ke halaman maindrill apakah value SPF dan Mandrill sudah di verifikasi.

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

Gambar 1.16 Hasil Verifikasi Domain di Mandrillapp

Gambar di atas merupakan gambar dari daftar *domain* yang terdaftar. Ketika pengguna ingin melakukan tes *domain* dapat menekan tombol test DNS settings. Nanti akan terlihat di bagian kolom DKIM setting dan SPF setting yang memperlihatkan apakah DKIM dan SPF telah valid atau error.

* 1. AWS Simple Email Service

Agar *engine* AWS SES dapat digunakan, beberapa konfigurasi harus dilakukan. Berikut merupakan konfigurasi yang harus dilakukan beserta penjelasan bagaimana cara melakukan konfigurasi tersebut

* + 1. Membuat AWS Credential

AWS Credential merupakan istilah yang digunakan untuk API *key* yang dimiliki AWS. Untuk membuat AWS Credential, anda dapat menuju *console* AWS dan memilih menu IAM.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.17 Dashboard AWS IAM

Gambar di atas merupakan gambar dari *dashboard* AWS Identity and Access Management (IAM). Untuk menambahkan *user*, anda dapat menekan tombol Add User dan mengikuti arahan yang diberikan oleh AWS. Hal ini mempermudah pengguna awam karena dapat menyelesaikan registrasi *user* IAM tanpa kesulitan. Setelah selesai menambahkan AWS IAM *user*, pengguna akan diberikan oleh AWS sebuah *file* yang memuat *access key* dan *secret access key*. *Access key* dan *secret access key* digunakan dalam menulis kode program nantinya. Untuk *credential* yang telah diberikan dapat ditaruh di ‘/root/.aws/’ dengan nama *file* “credential” pada penyimpanan *hosting*.

* + 1. Konfigurasi Domain

Setelah mendapatkan API *key*, pengguna harus melakukan konfigurasi *domain* yang digunakan untuk mengirimkan *email*. Pertama, pada *dashboard* AWS Simple Email Service, anda perlu menuju ke menu domains yang ada di *sidebar*. Setelah itu, anda dapat menambahkan *domain* untuk digunakan dalam mengirimkan *email*.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.18 Menu Domain pada AWS SES

Gambar di atas merupakan gambar dari menu Domains yang ditekan sebelumnya, Pengguna dapat langsung mengisi alamat *domain* yang digunakan untuk mengirim *email*.

Setelah pengguna mengisi alamat *domain* yang digunakan untuk mengirim *email,* pengguna harus melakukan verifikasi dan konfigurasi DKIM yang nantinya akan dikonfigurasi di CPanel *domain* tersebut. Untuk konfigurasi DKIM dan verifikasi dapat diunduh seperti contoh berikut

|  |
| --- |
| Graphical user interface, text, application, email  Description automatically generated Graphical user interface, text, application, email  Description automatically generated |

Gambar 1.19 Konfigurasi pada AWS SES

Gambar di atas merupakan gambar dari halaman detail domain yang telah didaftarkan. Anda dapat mengunduh konfigurasi DKIM dan verifikasi dengan menekan tulisan “Download Record Set as CSV”. Ketika sudah mendapatkan konfigurasi DKIM dan verifikasi tersebut, anda dapat menambahkan *file* TXT pada *hosting* yang telah anda miliki sebelumnya.

* + 1. Konfigurasi SPF dan DKIM Pada Domain

Ketika value SPF dan DKIM sudah didapatkan, sama seperti sebelumnya, kita dapat melakukan konfigurasi dengan cara memasukkan value yang sudah diberikan sebelumnya ke dalam DNS domain kita dengan cara, Login ke Alibaba cloud kemudian masuk ke halaman *console* dan klik menu pojok kiri atas dan pergi ke halaman Alibaba Cloud DNS lalu pilih domain yang didaftarkan sebelumnya pada AWS.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.20 DNS Setting pada Domain

Gambar di atas merupakan tampilan list Dns setting yang terdapat pada domain utama dari subdomain yang didaftarkan sebelumnya. Disini mengapa kita memilih domain yang agak berbeda dengan yang didaftarkan dikarenakan domain yang di daftarkan merupakan sub domain, maka dari itu pengaturan dnsnya tetap dilakukan melalui domain utama. Setelah halaman tersebut muncul, cara untuk menambahkan konfigurasi SPF dan DKIM dapat dengan menekan tombol *Add Record* lalu akan muncul sebuah form untuk menambahkan value baru.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |

Gambar 1.21 Konfigurasi DKIM dari AWS SES pada Domain

Gambar di atas merupakan value DKIM dari AWS SES. Sama seperti sebelumnya di Mandrill, kita dapat menambahkan value baru pada DNS setting dengan cara melakukan klik pada tombol *Add Record*. Setelah melakukan klik, maka akan mucnul form untuk menambahkan value baru dan kilk Confirm bila value sudah dirasa benar untuk menimpan value tersebut. Setelah DKIM selesai di masukkan maka kita dapat melanjutkan ke Langkah berikutnya yaitu menambahkan SPF.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |

Gambar 1.22 Konfigurasi SPF AWS SES pada Domain

Gambar di atas merupakan tampilan penambahan value dari konfigurasi SPF AWS SES. Dapat dilihat cara menambahkan SPF hampir sama dengan cara menambahkan DKIM hanya saja value dan *host*nya berbeda. Ketika keduanya sudah di tambahkan maka 2 value yang baru saja kita masukkan tadi akan otomatis terlihat pada list setting dns.

|  |
| --- |
| Text  Description automatically generated |

Gambar 1.23 List Value DKIM dan SPF AWS SES pada Domain

Gambar di atas merupakan tampilan list yang sudah berhasil dimasukkan value SPF dan DKIM dari AWS SES. Setelah list tersebut muncul, user dapat melakukan pengecekan Kembali ke halaman AWS SES apakah value SPF dan DKIM sudah di verifikasi.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.24 Pengecekan Konfigurasi Domain pada AWS SES

Gambar di atas merupakan gambar dari daftar *domain* yang telah terverifikasi dan telah dikonfigurasi. Proses pengecekan akan berlangsung secara otomatis dan ketika pengguna telah selesai menempatkan konfigurasi dan verifikasi pada *hosting*, maka status akan tertulis “Verified”.

* + 1. Konfigurasi Alamat Email

Alamat *email* yang digunakan untuk mengirim *email* harus diverifikasi terlebih dahulu. Hal ini sudah merupakan *rules* dari AWS dalam mengirim *email*. Pengguna dapat memilih menu “Email Addresses” yang terdapat pada *sidebar dashboard console* dari AWS SES.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.25 Daftar Email yang Terdaftar pada AWS SES

Gambar di atas merupakan tampilan menu Email Addresses. Di sini pengguna dapat menambahkan *email* yang digunakan untuk mengirim *email* kepada penerima. Karena status dari AWS SES yang digunakan masihlah berbentuk *sandbox*, *email* penerima harus juga didaftarkan di dalam AWS SES ini. Hal ini hanya berlaku untuk *email* yang terdaftar diluar *domain*. Pengguna dapat menambahkan *email* melalui tombol Verify a New Email Address dan pengguna hanya perlu memasukan *email* pengirim atau penerima luar. Secara otomatis, AWS akan mengirimkan *email* untuk proses verifikasi dan ketika *email* telah diverifikasi, status akan ditampilkan sebagai *verified*.

* 1. Alibaba Direct-mail

Agar *engine* Alibaba Direct-mail dapat digunakan, beberapa konfigurasi harus dilakukan. Berikut merupakan konfigurasi yang harus dilakukan beserta penjelasan bagaimana cara melakukan konfigurasi tersebut.

* + 1. Membuat Alibabacloud Access Key

Alibabacloud Access Key merupakan istilah yang digunakan untuk API key yang dimiliki Alibabacloud. Untuk membuat Alibabacloud Access Key, anda dapat menuju *console* Alibabacloud dan memilih menu Access Key.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.26 Tampilan Dashboard Console Alibaba Cloud

Gambar di atas merupakan gambar dari *dashboard console* Alibaba cloud Access Key. Pengguna dapat langsung menambahkan Access Key dan Alibabacloud langsung akan menambahkan kunci tersebut secara otomatis. Diharap kunci tersebut disimpan dan nantinya akan digunakan untuk keperluan pengkodean.

* + 1. Konfigurasi Domain

Setelah mendapatkan API *key*, pengguna harus melakukan konfigurasi *domain* yang digunakan untuk mengirimkan *email*. Pertama, pada *dashboard* Alibabacloud Direct-mail, anda perlu menuju ke menu Email Domains yang ada di *sidebar menu*. Setelah itu, anda dapat menambahkan *domain* untuk digunakan dalam mengirimkan *email*.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.27 Menu Email Domains

Gambar di atas merupakan gambar dari menu Email Domains yang ditekan sebelumnya, Pengguna dapat langsung menambahkan alamat *domain* yang digunakan untuk mengirim *email* dengan menekan New Domain.

Setelah pengguna mengisi alamat *domain* yang digunakan untuk mengirim *email,* pengguna harus melakukan verifikasi kepemilikan, verifikasi MX *record*, verifikasi CNAME *record*, dan konfigurasi SPF yang nantinya akan dikonfigurasi di CPanel *domain* tersebut. Untuk konfigurasi dan verifikasi dapat dilihat dengan cara pengguna menekan menu *configure* pada daftar *domain* yang telah didaftarkan pada Alibabacloud.

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

Gambar 1.28 Halaman Detail Domain

Gambar di atas merupakan gambar dari halaman detail domain yang telah didaftarkan. Anda dapat menyalin konfigurasi dan verifikasi yang diberikan oleh Alibabacloud pada *domain* yang telah didaftarkan. Ketika verifikasi dan konfigurasi telah ditambahkan di dalam *hosting*, maka status akan berubah menjadi *verification successfull*.

* + 1. Konfigurasi SPF dan DKIM Pada Domain

Sama seperti sebelumnya, cara menambahkan value SPF dan DKIM dapat dilakukan dengan cara memasukkan value yang sudah diberikan sebelumnya ke dalam DNS domain kita dengan cara, Login ke Alibaba cloud kemudian masuk ke halaman *console* dan klik menu pojok kiri atas dan pergi ke halaman Alibaba Cloud DNS lalu pilih domain yang didaftarkan sebelumnya pada AWS.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.29 Tampilan List DNS

Gambar di atas merupakan tampilan list Dns setting yang terdapat pada domain utama dari subdomain yang didaftarkan sebelumnya. Disini mengapa kita memilih domain yang agak berbeda dengan yang didaftarkan dikarenakan domain yang di daftarkan merupakan sub domain, maka dari itu pengaturan dnsnya tetap dilakukan melalui domain utama. Setelah halaman tersebut muncul, cara untuk menambahkan konfigurasi SPF dan DKIM dapat dengan menekan tombol *Add Record* lalu akan muncul sebuah form untuk menambahkan value baru.

|  |
| --- |
| A picture containing text, screenshot, monitor, computer  Description automatically generated |

Gambar 1.30 Konfigurasi DKIM Alibaba Direct Mail pada Domain

Gambar di atas merupakan value DKIM dari Alibaba Direct Mail. Sama seperti sebelumnya di AWS SES, kita dapat menambahkan value baru pada DNS setting dengan cara melakukan klik pada tombol *Add Record*. Setelah melakukan klik, maka akan muncul form untuk menambahkan value baru dan kilk Confirm bila value sudah dirasa benar untuk menimpan value tersebut. Setelah DKIM selesai di masukkan maka kita dapat melanjutkan ke Langkah berikutnya yaitu menambahkan SPF.

|  |
| --- |
| A picture containing text, screenshot, monitor, computer  Description automatically generated |

Gambar 1.31 Konfigurasi SPF Alibaba Direct Mail pada Domain

Gambar di atas merupakan tampilan penambahan value dari konfigurasi SPF Direct Mail. Dapat dilihat cara menambahkan SPF hampir sama dengan cara menambahkan DKIM hanya saja value dan *host*nya berbeda. Ketika keduanya sudah di tambahkan maka 2 value yang baru saja kita masukkan tadi akan otomatis terlihat pada list setting dns.

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

Gambar 1.32 List Value DKIM dan SPF Alibaba Direct Mail pada Domain

Gambar di atas merupakan tampilan list yang sudah berhasil dimasukkan value SPF dan DKIM dari Direct Mail. Setelah list tersebut muncul, user dapat melakukan pengecekan Kembali ke halaman Direct Mail untuk melakukan pengecekan apakah value SPF dan DKIM sudah di verifikasi.

* + 1. Konfigurasi Alamat Email

Alamat *email* yang digunakan untuk mengirim *email* harus diverifikasi terlebih dahulu. Hal ini sudah merupakan *rules* dari Alibabacloud dalam mengirim *email*. Pengguna dapat memilih menu “Sender Addresses” yang terdapat pada *sidebar dashboard console* dari Alibabacloud Direct-mail.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.33 Menu Email Address

Gambar di atas merupakan tampilan menu Email Addresses. Di sini pengguna dapat menambahkan *email* yang digunakan untuk mengirim *email* kepada penerima. Pengguna dapat menambahkan *email* melalui tombol Create Sender Address dan pengguna hanya perlu memasukan *email* pengirim. Pengguna dapat menekan tombol Verify the reply-to-address agar penerima dapat membalas *email* yang telah dikirimkan. Secara teknis tidak memerlukan verifikasi seperti AWS atau Mailchimp.

### Pengunduhan SDK

SDK (*software development kit*) merupakan sebuah *set tools* yang berisikan fungsi-fungsi untuk mengerjakan suatu proses. SDK dalam *engine email service* berguna untuk mengirimkan email secara otomatis. Mailchimp, AWS, dan Alibabacloud telah menyediakan *set* SDK untuk bahasa pemrograman PHP. Berikut merupakan alamat *website* untuk mengunduh *set* SDK dari Mailchimp, AWS, dan Alibaba.

|  |
| --- |
| //Mailchimp  https://github.com/mailchimp/mailchimp-transactional-php  //AWS  https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.zip  //Alibabacloud  http://aliyundm.oss-cn-hangzhou.aliyuncs.com/example/aliyun-php-sdk-dmV3.1.1.zip?spm=a2c63.p38356.879954.5.5a6d2b115JWm0W&file=aliyun-php-sdk-dmV3.1.1.zip |

Kode Program 1.2 Alamat Pengunduhan SDK

Kode program di atas merupakan link *website* untuk mengunduh SDK yang telah disediakan oleh Mailchimp, AWS, dan Alibabacloud. Setelah pengguna mengunduh SDK ini, pengguna dapat menaruh SDK ini di satu tempat dimana *script* yang digunakan untuk mengirim *email* diletakkan.

### Pengkodean

Setelah mengunduh SDK, pengguna dapat membuatkan sebuah *script* untuk mengirimkan *email* dari ketiga *service* tersebut. Secara garis besar, alur dari kode program yang akan dibuat meliputi pengecekan data yang akan digunakan untuk mengirim *email* di *database*, menghubungkan API masing-masing *service* melalui perantara SDK, mengirim *email* bedasarkan data yang ada di *database*, dan mengirimkan respon. Berikut merupakan kode program dari masing-masing *service* yang digunakan.

|  |
| --- |
| <?php  require\_once('vendor/autoload.php');  function run($message)  {  try {  $mailchimp = new MailchimpTransactional\ApiClient();  $mailchimp->setApiKey('uPXnD6d-hLn619S\_OUHO8A');  $mailchimp->messages->send(["message" => $message]);  return 1;  } catch (Error $e) {  return 2;  }  }  $connect = mysqli\_connect("menkz.xyz","notif\_basic1","notif") or die(mysqli\_error());  mysqli\_select\_db($connect,"notif\_basic1") or die("no database");  $check\_email = mysqli\_query($connect, "SELECT\*FROM outbox\_mail\_mailchimp WHERE flag = 1");  while($fetch = mysqli\_fetch\_array($check\_email))  {  $no = 0;  $message = [  "from\_email" => "testmailchimp@mailchimp.menkz.xyz",  "subject" => "Nontification From Our Service",  "text" => $fetch['out\_msg'],  "to" => [  [  "email" => $fetch['chat\_id'],  "type" => "to"  ]  ]  ];  $mailchimp = new MailchimpTransactional\ApiClient();  $mailchimp->setApiKey('uPXnD6d-hLn619S\_OUHO8A');  $response = $mailchimp->messages->send(["message" => $message]);  if($response[$no]->status == "sent")  {  mysqli\_query($connect, "UPDATE outbox\_mail\_mailchimp SET flag = 2 WHERE id\_outbox = $fetch[id\_outbox]");  echo "ok success";  }  else  {  echo "failed";  }  $no++;  }  ?> |

Kode Program 1.3 Kode Engine Mailchimp

Kode program di atas merupakan kode program yang digunakan untuk mengirim *email* via Mailchimp. Alamat *email* yang digunakan untuk mengirim *email* menggunakan alamat *email* yang telah terdaftar. Jika *email* sukses terkirim, maka *script* akan mengeluarkan respon “ok success”. Namun jika gagal, maka *script* akan mengeluarkan respon “Failed”.

|  |
| --- |
| <?php  // If necessary, modify the path in the require statement below to refer to the  // location of your Composer autoload.php file.  require 'vendor/autoload.php';  use Aws\Ses\SesClient;  use Aws\S3\S3Client;  use Aws\Exception\AwsException;  // Create an SesClient. Change the value of the region parameter if you're  // using an AWS Region other than US West (Oregon). Change the value of the  // profile parameter if you want to use a profile in your credentials file  // other than the default.  $SesClient = new SesClient([  'profile' => 'default',  'version' => 'latest',  'region' => 'ap-southeast-1',  ]);  $connect = mysqli\_connect("menkz.xyz","notif\_basic1","notif") or die(mysqli\_error());  mysqli\_select\_db($connect,"notif\_basic1") or die("no database");  $check\_email = mysqli\_query($connect, "SELECT\*FROM outbox\_mail\_aws WHERE flag = 1");  while($fetch = mysqli\_fetch\_array($check\_email))  {  // Replace sender@example.com with your "From" address.  // This address must be verified with Amazon SES.  $sender\_email = 'mangwahyu@menkz.xyz';  // Replace these sample addresses with the addresses of your recipients. If  // your account is still in the sandbox, these addresses must be verified.  $recipient\_emails = [$fetch['chat\_id']];  // Specify a configuration set. If you do not want to use a configuration  // set, comment the following variable, and the  // 'ConfigurationSetName' => $configuration\_set argument below.  $configuration\_set = '';  $subject = 'Nontification From Our Service';  $plaintext\_body = $fetch['out\_msg'] ;  $html\_body = $fetch['out\_msg'];  $char\_set = 'UTF-8';  try {  $email = [  'Destination' => [  'ToAddresses' => $recipient\_emails,  ],  'ReplyToAddresses' => [$sender\_email],  'Source' => $sender\_email,  'Message' => [  'Body' => [  'Html' => [  'Charset' => $char\_set,  'Data' => $html\_body,  ],  'Text' => [  'Charset' => $char\_set,  'Data' => $plaintext\_body,  ],  ],  'Subject' => [  'Charset' => $char\_set,  'Data' => $subject,  ],  ],  // If you aren't using a configuration set, comment or delete the  // following line  'ConfigurationSetName' => $configuration\_set,  ];  $result = $SesClient->sendEmail($email);  if ($SesClient->sendEmail($email))  {  mysqli\_query($connect, "UPDATE outbox\_mail\_aws SET flag = 2 WHERE id\_outbox = $fetch[id\_outbox]");  }  } catch (AwsException $e) {  // output error message if fails  echo $e->getMessage();  echo("The email was not sent. Error message: ".$e->getAwsErrorMessage()."\n");  echo "\n";  }  }  ?> |

Kode Program 1.4 Kode Engine AWS

Kode program di atas merupakan kode program yang digunakan untuk mengirim *email* via AWS SES. Alamat *email* yang digunakan untuk mengirim *email* menggunakan alamat *email* yang telah terdaftar. Jika *email* sukses terkirim, maka *script* tidak akan mengeluarkan respon apa-apa. Namun jika gagal, maka *script* akan mengeluarkan respon kode *error* dan alasan dari gagalnya *email* terkirim.

|  |
| --- |
| <?php  include\_once 'aliyun-php-sdk-core/Config.php';  use Dm\Request\V20151123 as Dm;  //Need to set up the corresponding region name, such as east China 1 region is set to cn-hangzhou, Singapore region is set to ap-southeast-1, Australia region is set to ap-southeast-2.  $iClientProfile = DefaultProfile::getProfile("ap-southeast-1", "LTAI5tRczo6UsT2P42s6Nqbp", "Jy7LdL7pKS28ytebn626O2toCzxNcp");  $iClientProfile::addEndpoint("ap-southeast-1","ap-southeast-1","Dm","dm.ap-southeast-1.aliyuncs.com");  //$iClientProfile::addEndpoint("ap-southeast-2","ap-southeast-2","Dm","dm.ap-southeast-2.aliyuncs.com");  $connect = mysqli\_connect("menkz.xyz","notif\_basic1","notif") or die(mysqli\_error());  mysqli\_select\_db($connect,"notif\_basic1") or die("no database");  $check\_email = mysqli\_query($connect, "SELECT\*FROM outbox\_mail\_alibaba WHERE flag = 1");  while($fetch = mysqli\_fetch\_array($check\_email))  {  $client = new DefaultAcsClient($iClientProfile);  $request = new Dm\SingleSendMailRequest();  //Singapore or Australia region need to set the version of the SDK, east China 1 (hangzhou) do not need to be set.  //$request->setVersion("2017-06-22");  $request->setAccountName("integrasi1@ims.menkz.xyz");  $request->setFromAlias("");  $request->setAddressType(1);  $request->setTagName("");  $request->setReplyToAddress("true");  $request->setToAddress($fetch['chat\_id']);  $request->setSubject("test");  $request->setHtmlBody($fetch['out\_msg']);  try {  $response = $client->getAcsResponse($request);  print\_r($response);  mysqli\_query($connect, "UPDATE outbox\_mail\_alibaba SET flag = 2 WHERE id\_outbox = $fetch[id\_outbox]");  }  catch (ClientException $e) {  print\_r($e->getErrorCode());  print\_r($e->getErrorMessage());  }  catch (ServerException $e) {  print\_r($e->getErrorCode());  print\_r($e->getErrorMessage());  }  }  ?> |

Kode Program 1.5 Kode Engine Alibabacloud

Kode program di atas merupakan kode program yang digunakan untuk mengirim *email* via Alibabacloud Direct-mail. Alamat *email* yang digunakan untuk mengirim *email* menggunakan alamat *email* yang telah terdaftar. Jika *email* sukses terkirim, maka *script* tidak akan mengeluarkan respon apa-apa. Namun jika gagal, maka *script* akan mengeluarkan respon kode *error* dan alasan dari gagalnya *email* terkirim.

### Pembuatan GUI dari Engine Email Service

Pembuatan GUI dimaksudkan untuk memonitoring apakah sebuah notifikasi terkirim atau tidak. GUI dibuat menggunakan PHPRunner. Berikut merupakan tampilan dari GUI *engine email service*.

|  |
| --- |
|  |

Gambar 1.34 Tampilan Login

Gambar di atas merupakan tampilan *login form* pada *engine* notifikasi. Pengguna diharuskan untuk melakukan *login* untuk menjaga keamanan dari *engine* notifikasi. *Username* yang digunakan adalah admin dan *password* yang digunakan adalah admin12345.

|  |
| --- |
|  |

Gambar 1.35 Tampilan Daftar Outbox AWS

Gambar di atas merupakan gambar dari daftar notifikasi yang akan dikirim melalui *engine* AWS Simple Email Service. Ketika Flag menunjukan angka 2, maka notifikasi telah terkirim ke alamat yang dituju. Pengguna dapat menambahkan notifikasi baru dengan menekan tombol Add New.

|  |
| --- |
|  |

Gambar 1.36 Tampilan Daftar Outbox Alibaba

Gambar di atas merupakan gambar dari daftar notifikasi yang akan dikirim melalui *engine* Alibabacloud Direct-mail. Ketika Flag menunjukan angka 2, maka notifikasi telah terkirim ke alamat yang dituju. Pengguna dapat menambahkan notifikasi baru dengan menekan tombol Add New.

|  |
| --- |
|  |

Gambar 1.37 Tampilan Daftar Outbox Mailchimp

Gambar di atas merupakan gambar dari daftar notifikasi yang akan dikirim melalui *engine* Mailchimp. Ketika Flag menunjukan angka 2, maka notifikasi telah terkirim ke alamat yang dituju. Pengguna dapat menambahkan notifikasi baru dengan menekan tombol Add New.

### Instalasi AAPANEL

aaPanel adalah perangkat lunak control panel web hosting, mendukung distribusi Linux CentOS, Ubuntu, dan Debian. aaPanel menggunakan web server stack LAMP (Linux Apache MariaDB PHP) atau LNMP (Linux Nginx MariaDB PHP). aaPanel memakai lisensi open source dan bisa didapatkan dengan gratis. Berikut adalah cara instalasinya.

|  |
| --- |
|  |

Gambar 1.38 Login SSH ke VPS/Hosting

Gambar di atas merupakan tampilan Login SSH ke vps/hosting. Langkah pertama yang dilakukan dalam instalasi AAPANEL adalah melakukan login ke VPS/Hotsing dengan menggunakan SSH. Pada kasus ini kita menggunakan OS CentOS 8 dengan spesifikasi Ram 1gb dan 1core Cpu. Setelah login ke VPS langkah selanjutnya adalah memasukkan kode CLI untuk melakukan instalasi AAPANEL.

|  |
| --- |
| yum install -y wget && wget -O install.sh http://www.aapanel.com/script/install\_6.0\_en.sh && bash install.sh |

Kode Program 1.6 Kode CLI Instalasi AAPanel

Kode Program di atas merupakan kode CLI untuk melakukan instalasi Aapanel pada vps dengan OS CentOs. Setelah kode tersebut di jalankan, aka nada pertanyaan berupa “*Do you want to install aaPanel to the /www directory now?(y/n)”,* dan ketik Y lalu tekan enter untuk melanjutkan peng-installan, dan tunggu hingga proses instalasi sampai selesai.

|  |
| --- |
| Install aaPanel selesai |

Gambar 1.39 Proses Instalasi AAPanel

Gambar di atas merupakan tampilan ketika proses instalasi Aapanel telah selesai dan berhasil. Dapat dilihat, Aapanel akan memberikan kita url untuk login ke panel itu sendiri dan juga default user dari panel. Selain itu, Aapanel akan merekomendasikan untuk mengganti Port default untuk keamanan pengguna. Setelah instalasi selesai, kita dapat masuk ke link menggunakan user dan password default yang sudah diberikan sebelumnya.

Tabel 1.1 Default Akun AAPanel

|  |  |  |
| --- | --- | --- |
| Link Default | Username | Password |
| http://64.20.51.164:888/4c63f162 | 819ln96q | 031e6bb2 |

Tabel di atas merupakan default password yangdiberikan oleh AAPANEL saat pertama kali melakukan instalasi. User dapat mengakses panel dengan akun tersebuut untuk pertama kalianya. Default dari url tersebut berupa Ip public dari VPS yang nantinya dapat di ganti dengan alamat domain. Saat ini AApanel hanya support untuk 1 panel 1 user dan tidak dapat di tambahkan user lainnya di dalam satu panel.

|  |
| --- |
| Login aaPanel |

Gambar 1.40 Halaman Login Aapanel

Gambar di atas merupakan tampilan login dari Aapanel. Langkah selanjutnya setelah berhasil menginstalasi Aapanel adalah login ke Aaapanel dengan user dan password default yang di berikan lalu melakukan penginstalan web server. Setelah berhasil login pada halaman di atas, kita akan diarahkan ke halaman utama Aapanel.

|  |
| --- |
| aaPanel - Software Packages |

Gambar 1.41 Halaman Utama Aapanel

Gambar di atas merupakan tampilann halaman depan/utama dari Aapanel. Setelah login untuk pertama kalinya, Aapanel akan memberikan 2 pilihan pada user untuk menginstall webserver pada hostingan. Kedua pilihan tersebut antara lain adalah Nginx & Apache (lamp). Pada kasus kali ini kita akan menggunakan webserver Lampp atau apache dengan cara menekan tombol One-Click maka proses instalasi akan langsung berjalan.

|  |
| --- |
| Proses install web server |

Gambar 1.42 Tampilan Proses Instalasi Lampp

Gambar di atas merupakan proses penginstalan web server Lampp. Pengguna hanya perlu menunggu proses tersebut selesai dan ketika proses penginstalan selesai maka aka nada pesan baru yang muncul

|  |
| --- |
| Install web server selesai |

Gambar 1.43 Tampilan Proses Instalasi Lampp Sukses

Gambar di atas merupakan pesan bahwa proses instalasi Lampp berhasil dilakukan. Setelah pesan tersebut muncul, pengguna dapat menutup *message box* tersebut dan Aapanel sudah siap untuk digunakan untuk melakukan hosting website.

### Mengganti Url Default

Pada AApanel, default link untuk mengakses Panel adalah Ip public dari vps sementara port default dari AApanel adalah 8888, AApanel merekomendasikan kita untuk mengganti url tersebut untuk keamanan. Berikut penjelasannya.

|  |
| --- |
|  |

Gambar 1.44 Tampilan Halaman Setting AAPanel

Gambar di atas merupakan tampilan dari halaman setting AApanel. Untuk mengganti domain default AApanel, pertama klik menu Settings, sehingga kita akan di arahkan ke halaman setting AApanel. Setelah berada di halaman setting AApnael, kita dapat melihat form input post dan domain. Untuk melakukan penggantian default port, cukup dengan merubah port pada form Panel port, sementara untuk memasukkan domain kita agar menjadi URL, kiita dapat memasukkan domain ke form domain yang dalam kasus ini kita menggunakan domain “menkz.xyz”, setelah itu scroll ke bawah dan tekan tombol save maka link dan port default AApanel sudah berhasil di rubah. Setelah kita merubah domain default ke domain milik kita. Sekarang kita sudah dapat lanjut ke tahap selanjutnya yaitu memasang SSl pada panel.

### Memasang SSl

Setelah berhasil memasang AAPANEL dan mengganti default url, kita akan lanjut ke tahap berikutnya yaitu cara memasangkan SSL pada website yang terdapat pada AAPANEL. Berikut adalah penjelasannya.

|  |
| --- |
|  |

Gambar 1.45 Tampilan Halaman Menu Website

Gambar di atas merupakan tampilan halaman menu website. Halaman tersebut berisi list url website apa saja yang ada pada panel. Untuk mengisi SSl pada website, kita dapat mengklik salah satu webisite yang ada dan akan muncul menu pop-up seperti dibawah ini

|  |
| --- |
|  |

Gambar 1.46 Tampilan Menu Popup dari Website AAPanel

Gambar diatas merupakan tampilan menu popup dai website yang ada pada AApanel. Pada menu popup tersebut, klik menu SSL. Setelah menu SSl di klik, maka akan muncul beberapa tab di sebelah kanan atas. Pada tab tersebut terdapat 3 menu utama yaitu : Lets Encrypt, Other Certificate, dan Turn Of. Jika user ingin memasangkan key SSl milik sendiri, user dapat mengklik tab Other Certificate. Kemudian memasukkan Key yang dimiliki. Dalam kasus ini, kita akan menggunakan SSl Lets Encrypt yang sudah di sediakan oleh AAPANEL, dengan cara klik tab Let’s Encrypt dan pilih domain yang ingin di daftar kan seperti pada gambar di atas. Setelah domain di pilih, klik Apply dan proses SSL akan berjalan secara otomatis seperti di bawah ini.

|  |
| --- |
|  |

Gambar 1.47 Proses Instalasi SSI

Gambar di atas merupakan gambar proses instalasi SSl. Proses tersebut akan berjalan otomatis ketika tombal Apply di klik. Ketika berhasil dialog box tersebut akan otomatis ter tutup dan user dapat melakukan tes ke website yang di daftarkan tadi dengan cara menambahkan https di belakang domain.

### Hosting

Setelah sebelumnya kita sudah membuat *engine* dan *engine* tersebut sudah bekerja di local, kini saatnya kita meng-upload *engine* tersebut ke *hosting* yang kita memiliki agar dapat berjalan secara realtime. Hal pertama yang harus dilakukan adalah login ke halaman panel *hosting* yang kita miliki. Dalam kasus ini panel yang digunakan adalah AAPANEL.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |

Gambar 1.48 Tampilan halaman Utama Aapanel

Gambar di atas merupakan tampilan awal dari Panel *hosting* AAPANEL. Ketika sudah berhasil login dan masuk ke halaman tersebut, klik ke menu file dan kita akan di arahkan ke halaman file exploler dari *hosting*.

|  |
| --- |
| A picture containing text, screenshot, computer, computer  Description automatically generated |

Gambar 1.49 Tampilan Halaman File Explorer AAPanel

Gambar di atas merupakan tampilan halaman *file explorer* pada *hosting*. Ketika sudah berada di halaman tersebut, pertama kita harus membuat folder baru yang akan jadi tempat untuk meng upload *engine* yang sudah kita buat. Cara untuk membuat folder baru adalah, klik tombol new kemudian pilih New Directory seperti gambar di atas dan beri nama pada folder tersebut.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |

Gambar 1.50 Tampilan Upload File

Gambar di atas merupakan tampilan *upload file* pada *Hosting*. Setelah berhasil membuat folder, masuk ke folder yang baru saja dibuat, dan klik tombol Upload yang ada di pojok kiri atas dan akan muncul *pop up* seperti gambar di atas. Sebelum itu, pastikan file *engine* sudah di compress terlebih dahulu ke dalam format zip/rar. Langkah selanjutnya adalah pada pop-up tersebut, klik tombol upload file dan pilih file *engine* yang sudah di compress lalu klik tombol upload pada pojok kanan bawah.

|  |
| --- |
| Graphical user interface, text, application, email  Description automatically generated |

Gambar 1.51 Tampilan Menu Extract

Gambar di atas merupakan tampilan menu *extract* pada *hosting*. Jika file sudah ter-upload, klik kanan pada file rar/zip lalu pilih menu unzip untuk melakukan extract pada file tersebut. Jika file sudah berhasil diextract, maka *engine* sudah berhasil di upload ke *hosting* akan tetapi sebelum itu, kita perlu meng upload sebuah file Bernama “*credential*” agar *engine* AWS SES dapat berjalan dengan semestinya.

|  |
| --- |
| A screenshot of a computer  Description automatically generated |

Gambar 1.52 lokasi penempatan *file* “Credential”

Gambar di atas merupakan lokasi penempatan *file* “Credential”. File tersebut di letakkan pada directory “Root/” kemudian di dalam folder root dibuatkan Kembali folder baru bernama “.aws”. Setelah folder tersebut di buat, file “Credential” diletakan di dalam folder “.aws”. Jika file credential sudah terupload maka kita akan melakukan Implementasi Cronjob.

### Implementasi Cronjob Hosting

Cron adalah tool di sistem operasi berbasis UNIX (Linux, Ubuntu, dan lain-lain) yang berfungsi untuk menjalankan task atau script secara otomatis. Oleh karena itu, cron job adalah sebutan penggunaan cron untuk menjadwalkan task di waktu yang sudah ditentukan secara berulang kali.Pada penggunaannya, cron job tidak hanya dapat dimanfaatkan untuk backup, pemeliharaan server, dan keperluan teknis *hosting* saja. Salah satu contohnya kita dapat menjadwalkan sebuah script berjalan dengan rentang waktu tertentu. Berikut adalah Langkah Langkah implementasi cronjob pada *Engine* Email.

|  |
| --- |
| A picture containing text, screenshot, monitor, computer  Description automatically generated |

Gambar 1.53 Tampilan Halaman Cronjob

Gambar di atas merupakan tampilan halaman *cronjob* pada panel *hosting*. Langkah pertama untuk implmentasi *cronjob* setelah *engine* diupload adalah pergi ke menu Cron dan buat sebuah *cronjob* baru pada halaman *cronjob*. Pada halaman *cronjob* ada beberapa configurasi untuk membuat *cronjob*. Berikut adalah configurasnya.

|  |
| --- |
| A computer screen shot  Description automatically generated with low confidence |

Gambar 1.54 Konfigurasi Cronjob

Gambar di atas Merupakan salah satu contoh configurasi implementasi cronjob pada service Alibaba. Dapat dilihat configurasi yang diperlukan adalah, “*Type of Task*” yaitu tipe dari tugas/task yang di jalankan cronjob, dimana dalam kasus ini menggunakan tipe shell script yang nantinya perintahnya berupa kode CLI. Lalu kedua ada *Name of task* yang merupakan nama dari task yang kita buat, lalu yang ketiga ada *Execution Cycle*, dimana konfigurasi ini akan mengatur waktu running dari script/task yang kita buat, dan yang terakhir ada *Script Content*, dimana configurasi ini berisikan perintah CLI yang akan mengeksekusi script yang diupload sebelumnya.

Dapat di lihat, perintah tesebut akan mengeksekusi filescript yang berada pada directory www/wwwroot/Integrasi.menkz.xyz/Alibaba/test.php”. Perintah tersebut akan dieksekusi berulang setiap 1 menit sekali, dan untuk implementasi Cronjob untuk service yang lainnya terbilang sama persis dan hanya perlu merubah lokasi folder dari API masing-masing service.

Script untuk Mailchimp terletak pada riectory “/www/wwwroot/Integrasi.menkz.xyz/mailchimp/test.php” sementara untuk script AWS terletak di “php -q /www/wwwroot/Integrasi.menkz.xyz/AWS/test.php”. Seteleah seluruh configurasi sudah selesai diisi maka kita dapat melakukan klik pada tombol “Add task” untuk menyimpan cronjob yang sudah kita buat.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |

Gambar 1.55 List Cronjob yang Tersimpan

Gambar di atas merupakan list cronbjob yang tersimpan pada *hosting*. Dapat dilihat seluruh cronjob sudah dapat dimasukkan dengan benar dan sudah tersimpan pada *hostingan*. Pengguna dapat melakukan pengecekan apakah cronjob sudah berjalan atau tidak dengan cara meng-klik tombol log ataupun bisa menjalankan secara paksa dengan cara meng-klik tombol execute. Setelah berhasil mengimplementasi cronjob, kita akan ke tahap akhir yaitu Testing / pengujian.

## Pengujian Engine

Tahap terakhir adalah tahap pengujian *engine* pada Cronjob yang sudah diimplemntasi sebelumnya. Pada tahap tesing akan dilakukan pengetesan pada ketiga service tersebut apakah cronjob dan *engine* sudah berjalan dengan semestinya atau belum. Berikut adalah penjelasannya

### Pengujian Pada Service Mailchimp

Pengujian pertama dilakukan pada sevice mailchimp. Service mailchimp hanya dapat mengirimkan email ke domain yang sudah terverifikasi saja. Pada service mailchimp, domain yang sudah terverifikasi merupakan sub-domain dari menkz.xyz yaitu mailchimp.menkz.xyz sehingga dalam pengujian kita akan menggunakan email dengan domain tersebut baik untuk mengirim maupun menerima email.

|  |
| --- |
| A picture containing text, screenshot, computer, computer  Description automatically generated |

Gambar 1.56 List Data pada Tabel outbox\_mail\_mailchimp Sebelum Ditambah

Gambar di atas merupakan tampilan list data dari table outbox\_mail\_mailchimp yang terdapat pada database *hostingan*. Pengujian dilakukan dengan cara memasukan data email pada table outbox\_mail\_mailchimp secara manual, sehingga nantinya Ketika cronjob berjalan, maka email tersebut akan otomatis terkirim walaupun kita memasukkannya secara manual.

|  |
| --- |
| A picture containing text, screenshot, computer, computer  Description automatically generated |

Gambar 1.57 List Data pada Tabel outbox\_mail\_mailchimp Setelah Ditambah

Gambar di atas merupakan tampilan list data setelah ditambahkan data secara manual. Dapat dilihat flag dari data tersebut beri nilai “1” dimana artinya data tersebut belum terkirim dan data tersebut harusnya mengirimkan email ke email “”. Untuk mengetahui cronjob dan *engine* sudah berjalan atau belum, kita dapat pergi ke hoting lagi dan mengklik tombil log pada cronjob “Check Email Mailchimp”.

|  |
| --- |
| A picture containing text, screenshot, monitor, computer  Description automatically generated |

Gambar 1.58 Log Cronjob Check Email Mailchimp

Gambar di atas merupakan tampilan log dari cronjob “Check Email Mailchimp”. Pada gambar tersebut terlihat ada feedback berupa tulisan oke success yang berarti script sudah berhasil di jalankan oleh cronjob pada *hostingan* dan setelah terlihat pada log berhasil, kita dapat mengecek pada inbox email apakah email tersebut sudah masuk atau belum.

|  |
| --- |
| A screenshot of a computer  Description automatically generated with medium confidence |

Gambar 1.59 Email Berhasil Dikirim

Gambar di atas merupakan tampilan dari email yang sudah berhasil dikirim oleh service mailchimp. Dapat dilihat isi dari email tersebut sama dengan isi dari field ”out\_msg” pada table outbox\_mail\_mailchimp sebelumnya. Sekarang kita juga harus melakukan pengecekan, apakah pada database, flag dari data yang kita masukkan sudah berubah atau belum.

|  |
| --- |
| A picture containing text, screenshot, computer, computer  Description automatically generated |

Gambar 1.60 List Data pada Tabel outbox\_mail\_mailchimp Setelah Cronjob Berjalan

Gambar di atas merupakan tampilan list data dari database pada *hostingan*. Dapat dilihat flag dari data yang tadi kita masukkan secara manual sudah berubah ke 2 dimana artinya pesan sudah berhasil dikirim dan cronwjob setra *engine* Mailchimp sudah bekerja dengan semestinya.

### Pengujian Pada Service AWS

Pengujian kedua dilakukan pada sevice AWS. Service AWS dapat mengirimkan email ke email dengan domain mana saja walaupun domain berbeda dengan domain yang sudah terverifikasi, akan tetapi email tersebut harus sudah terverifikasi pada *console* AWS SES dimana ini agak berbeda dengan mailchimp yang membebaskan kita mengirim email pada email manapun tanpa perlu verifikasi selama email tersebut memiliki domain yang sama seperti domain sudah terverifikasi. Hal ini dapat diperbaiki dengan keluar dari mode SandBox akan tetapi dikarenakan berbayar, dalam pengujian kali ini kita akan menggunakan versi gratisnya. Pada service AWS, domain yang sudah terverifikasi merupakan domain utama yaitu menkz.xyz.

|  |
| --- |
| A computer screen shot  Description automatically generated with low confidence |

Gambar 1.61 List Data pada Tabel outbox\_mail\_aws Sebelum Ditambah

Gambar di atas merupakan tampilan list data dari table outbox\_mail\_aws yang terdapat pada database *hostingan*. Pengujian dilakukan dengan cara memasukan data email pada table outbox\_mail\_aws secara manual, sehingga nantinya ketika cronjob berjalan, maka email tersebut akan otomatis terkirim walaupun kita memasukkannya secara manual.

|  |
| --- |
| A computer screen shot  Description automatically generated with low confidence |

Gambar 1.62 List Data pada Tabel outbox\_mail\_aws Setelah Ditambah

Gambar di atas merupakan tampilan list data setelah ditambahkan data secara manual. Dapat dilihat flag dari data tersebut beri nilai “1” dimana artinya data tersebut belum terkirim dan data tersebut harusnya mengirimkan email ke email [pandu.pratama871@gmail.com](mailto:pandu.pratama871@gmail.com) dimana kita dapat mengirimkan email ke email tersebut walaupun memiliki domain yang berbeda, dikarenakan email tersebut sudah terverifikasi pada *console* AWS SES. Untuk mengetahui cronjob dan *engine* sudah berjalan atau belum, kita dapat pergi ke hoting lagi dan mengklik tombil log pada cronjob “Check Email AWS”.

|  |
| --- |
| A picture containing text, screenshot, computer, monitor  Description automatically generated |

Gambar 1.63 Log Cronjob Check Email AWS

Gambar di atas merupakan tampilan log dari cronjob “Check Email AWS”. Pada gambar tersebut dapat terlihat keterangan success yang berarti script sudah berhasil di jalankan oleh cronjob pada *hostingan*. Pada gambar tesebut mengapa tidak terlihat feedback apapun selain success dikarenakan pada script dari AWS sendiri, jika berhasil, maka tidak akan memebrikan feedback apapun, namu sebaliknya jika error maka akan memberikan feedback error berupa kode error. Setelah feedback tersebut muncul, kita dapat mengecek pada inbox email apakah email tersebut sudah masuk atau belum.

|  |
| --- |
| Text  Description automatically generated with medium confidence |

Gambar 1.64 Email Berhasil Dikirim

Gambar di atas merupakan tampilan dari email yang sudah berhasil dikirim oleh service AWS. Dapat dilihat isi dari email tersebut sama dengan isi dari field ”out\_msg” pada table outbox\_mail\_aws sebelumnya. Sekarang kita juga harus melakukan pengecekan, apakah pada database, flag dari data yang kita masukkan sudah berubah atau belum.

|  |
| --- |
| A computer screen shot  Description automatically generated with low confidence |

Gambar 1.65 List Data pada Tabel outbox\_mail\_aws Setelah Cronjob Berjalan

Gambar di atas merupakan tampilan list data dari database pada *hostingan*. Dapat dilihat flag dari data yang tadi kita masukkan secara manual sudah berubah ke 2 dimana artinya pesan sudah berhasil dikirim dan cronjob serta *engine* AWS sudah bekerja dengan semestinya.

### Pengujian Pada Service Alibaba

Pengujian terakhir dilakukan pada sevice Alibaba. Service Alibaba dapat mengirimkan email ke email dengan domain mana saja dan tidak terikat seperti halnya AWS yang harus terverifikasi terlebih dahulu, akan tetapi email yang kita kirimkan akan masuk ke folder SPAM. Pada service Alibaba, domain yang sudah terverifikasi merupakan sub-domain dari menkz.xyz yaitu ims. menkz.xyz.

|  |
| --- |
| Graphical user interface, text, application  Description automatically generated |

Gambar 1.66 List Data pada Tabel outbox\_mail\_alibaba Sebelum Ditambah

Gambar di atas merupakan tampilan list data dari table outbox\_mail\_alibaba yang terdapat pada database *hostingan*. Pengujian dilakukan dengan cara memasukan data email pada table outbox\_mail\_alibaba secara manual, sehingga nantinya ketika cronjob berjalan, maka email tersebut akan otomatis terkirim walaupun kita memasukkannya secara manual.

|  |
| --- |
| A picture containing text, screenshot, computer, computer  Description automatically generated |

Gambar 1.67 List Data pada Tabel outbox\_mail\_alibaba Setelah Ditambah

Gambar di atas merupakan tampilan list data setelah ditambahkan data secara manual. Dapat dilihat flag dari data tersebut beri nilai “1” dimana artinya data tersebut belum terkirim dan data tersebut harusnya mengirimkan email ke email [komangpermana7@gmail.com](mailto:komangpermana7@gmail.com) dimana kita dapat mengirimkan email ke email tersebut walaupun memiliki domain yang berbeda, akantetapi email tersebut akan masuk ke folder spam. Untuk mengetahui cronjob dan *engine* sudah berjalan atau belum, kita dapat pergi ke hoting lagi dan mengklik tombil log pada cronjob “Check Email Alibaba”.

|  |
| --- |
| A picture containing text, screenshot, computer, monitor  Description automatically generated |

Gambar 1.68 Log Cronjob Check Email Alibaba

Gambar di atas merupakan tampilan log dari cronjob “Check Email Alibaba”. Pada gambar tersebut dapat terlihat keterangan success serta feedback berupa requestId dan Env Id yang berarti script sudah berhasil di jalankan oleh cronjob pada *hostingan*. Setelah feedback tersebut muncul, kita dapat mengecek pada inbox email apakah email tersebut sudah masuk atau belum.

|  |
| --- |
| Text  Description automatically generated |

Gambar 1.69 Email Berhasil Dikirim

Gambar di atas merupakan tampilan dari email yang sudah berhasil dikirim oleh service Alibaba. Dapat dilihat isi dari email tersebut sama dengan isi dari field ”out\_msg” pada table outbox\_mail\_alibaba sebelumnya dan dapat dilihat email tersebut masuk ke folder SPAM. Sekarang kita juga harus melakukan pengecekan, apakah pada database, flag dari data yang kita masukkan sudah berubah atau belum.

|  |
| --- |
| A picture containing text, screenshot, computer, computer  Description automatically generated |

Gambar 1.70 List Data pada Tabel outbox\_mail\_alibaba Setelah Cronjob Berjalan

Gambar di atas merupakan tampilan list data dari database pada *hostingan*. Dapat dilihat flag dari data yang tadi kita masukkan secara manual sudah berubah ke 2 dimana artinya pesan sudah berhasil dikirim dan cronjob serta *engine* Alibaba sudah bekerja dengan semestinya.

# SISTEM DAN WEBVIEW PHPRUNNER

## Tahapan Pembuatan Sistem Pada PHPRunner

### Manajemen Akun Personal

|  |
| --- |
|  |

Gambar 2. 1 Pengaturan Login

Gambar 2.1 merupakan pengaturan dari login sistem. Login pada sistem mengacu kepada tabel personal dengan username field yaitu mail, password field yaitu password dan full name field yaitu name. Ketika user melakukan logi akan disesuaikan dengan data yang ada di tabel personal.

|  |
| --- |
|  |

Gambar 2. 2 Pengaturan Registrasi

Gambar 2.2 merupakan pengaturan dari registrasi sistem. Ketika User melakukan registrasi user akan dikirim email yang berisikan link aktivasi akun ke alamat email yang diinputkan user pada form registrasi. Password yang diinputkan user juga akan di enkripsi dengan BYCRYPT Algorithm untuk keamanan data user.

|  |
| --- |
| $\_SESSION["member\_id"] = $data["member\_id"];  $data = array();  $data["user\_type"] = 2;  $data["member\_id"] = $\_SESSION["member\_id"];  $rs = DB::Select("group\_member", $data );  while( $record = $rs->fetchAssoc() )  {  $\_SESSION["group\_id"] = $record["group\_id"];  }  // Place event code here.  // Use "Add Action" button to add code snippets. |

Kode Program 2. 1 After Successful Login

Kode program 2.1 adalah kode program yang berada di event after successful login. Kode program ini berfungsi untuk mendeklarasi dan inisialisasi dua buah variabel session yaitu member\_id dan group\_id. Session member\_id didapatkan dari value member\_id dari user yang login.

### Manajemen Agenda Grup

Agenda grup adalah agenda yang dibuat oleh admin grup. Agenda grup akan disebar kepada member grup yang melakukan subscriber pada jenis agenda tertentu.

|  |
| --- |
|  |

Gambar 2. 3 Pengaturan Dropdown grup

Gambar 2.3 merupakan pengaturan untuk daftar grup pada dropdown. Daftar grup yang akan ditampilkan hanya grup yang dimiliki oleh admin tersebut.

|  |
| --- |
|  |

Gambar 2. 4 Pengaturan Dropdown tipe agenda grup

Gambar 2.4 merupakan pengaturan untuk daftar tipe agenda grup pada dropdown. Daftar tipe agenda grup yang akan ditampilkan hanya tipe agenda grup yang dimiliki oleh grup tersebut.

|  |
| --- |
| $idm = $values["group\_agenda\_id"];  if($values['loop\_id'] == 2)  {  $data = array();  $keyvalues = array();  $data["loop\_value"] = NOW();  $keyvalues["group\_agenda\_id"] = $values["group\_agenda\_id"];  DB::Update("group\_agenda", $data , $keyvalues);  }  if($values["message\_general"] == "General")  {  $data = array();  $data["group\_id"] = $values[group\_id];  $rs = DB::Select("group\_member", $data );  while( $record = $rs->fetchAssoc() )  {  $mmid = $record["member\_id"];  $gmid = $record["group\_member\_id"];  $agendaType = $values["agenda\_type\_id"];  $rs1 = DB::Query("select member\_id from group\_member\_agenda\_subscriber  where agenda\_type\_id = '$agendaType' AND member\_id = '$mmid'");  $dataAgendaType = $rs1->fetchAssoc();  if($dataAgendaType)  {  $data2 = array();  $data2["group\_member\_id"] = $gmid;  $data2["member\_id"] = $mmid;  $data2["group\_agenda\_id"] = $values['group\_agenda\_id'];  DB::Insert("group\_member\_agenda", $data2 );  $tgl = $values["loop\_value"];  $repeat = $values["repeat"];  if($repeat == 0)  {  $data4 = array();  $data4["member\_id"] = $mmid;  $data4["id\_group\_agenda"] = $values["group\_agenda\_id"];  $data4["date\_send"] = $values['loop\_value'];  DB::Insert("group\_agenda\_general\_logs", $data4 );  $data3 = array();  $data3["out\_msg"] = $values["message\_content"];  $data3["type"] = "msg";  $data3["flag"] = 1;  $data3["tgl"] = $values["loop\_value"];  $rs2 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 1");  $dataChannelTele=$rs2->fetchAssoc();  if($dataChannelTele)  {  $data3["chat\_id"] = $dataChannelTele["chat\_id"];  DB::Insert("outbox\_telegram", $data3 );  }  $rs3 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 2");  $dataChannelAws=$rs3->fetchAssoc();  if($dataChannelAws)  {  $data3["chat\_id"] = $dataChannelAws["chat\_id"];  DB::Insert("outbox\_mail\_aws", $data3 );  }  $rs4 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 3");  $dataChannelWa=$rs4->fetchAssoc();  if($dataChannelWa)  {  $data3["chat\_id"] = $dataChannelWa["chat\_id"];  DB::Insert("outbox\_whatsapp", $data3 );  }  $rs5 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 4");  $dataChannelLine=$rs5->fetchAssoc();  if($dataChannelLine)  {  $data3["chat\_id"] = $dataChannelLine["chat\_id"];  DB::Insert("outbox\_line", $data3 );  }  }  else  {  $data3 = array();  $i = 0;  while($i <= $repeat)  {  $rep = $values["repeat\_type"];  if($rep == "bydate")  {  $data1 = array();  $data1["out\_msg"] = $values["message\_content"];  $data1["type"] = "msg";  $data1["flag"] = 1;  $data2 = array();  $data4 = array();  $data4["member\_id"] = $mmid;  $data4["id\_group\_agenda"] = $values["group\_agenda\_id"];  $data4["date\_send"] = $tgl;  if($i == 0)  {  $data1["tgl"] = $tgl;  $rs1 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 1");  $dataChannelTele=$rs1->fetchAssoc();  if($dataChannelTele)  {  $data1["chat\_id"] = $dataChannelTele["chat\_id"];  DB::Insert("outbox\_telegram", $data1 );  }  $rs2 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 2");  $dataChannelAws=$rs2->fetchAssoc();  if($dataChannelAws)  {  $data1["chat\_id"] = $dataChannelAws["chat\_id"];  DB::Insert("outbox\_mail\_aws", $data1 );  }  $rs3 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 3");  $dataChannelWa=$rs3->fetchAssoc();  if($dataChannelWa)  {  $data1["chat\_id"] = $dataChannelWa["chat\_id"];  DB::Insert("outbox\_whatsapp", $data1 );  }    $rs4 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 4");  $dataChannelLine=$rs4->fetchAssoc();  if($dataChannelLine)  {  $data1["chat\_id"] = $dataChannelLine["chat\_id"];  DB::Insert("outbox\_line", $data1 );  } DB::Insert("group\_agenda\_general\_logs", $data4 );  }  else  {  //$data2["group\_agenda\_id"] = $idm;  //$data2["dodate"] = now();  //DB::Insert("group\_agenda\_repeat\_date", $data2 );  //DB::Insert("group\_agenda\_general\_logs", $data4 );  }  }  else  {  if($i == 0)  {  $data3["tgl"] = $tgl;  $tgl3 = $tgl;  }else if($i > 0)  {  if($values["repeat\_type"] == "daily")  {  $tgl2 = $data3["tgl"];  $tgl3 = date('Y-m-d H:i:s', strtotime($tgl2 . ' +1 day' ));  $data3["tgl"] = $tgl3;  }else if($rep == "weekly")  {  $tgl2 = $data3["tgl"];  $tgl3 = date('Y-m-d H:i:s', strtotime($tgl2 . ' +1 week' ));  $data3["tgl"] = $tgl3;  }else if($rep == "monthly")  {  $tgl2 = $data3["tgl"];  $tgl3 = date('Y-m-d H:i:s', strtotime($tgl2 . ' +1 month' ));  $data3["tgl"] = $tgl3;  }else if($rep == "yearly")  {  $tgl2 = $data3["tgl"];  $tgl3 = date('Y-m-d H:i:s', strtotime($tgl2 . ' +1 year' ));  $data3["tgl"] = $tgl3;  }  }  $data4 = array();  $data4["member\_id"] = $mmid;  $data4["id\_group\_agenda"] = $values["group\_agenda\_id"];  $data4["date\_send"] = $tgl3;  DB::Insert("group\_agenda\_general\_logs", $data4 );      $data3["out\_msg"] = $values["message\_content"];  $data3["type"] = "msg";  $data3["flag"] = 1;  $rs2 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 1");  $dataChannelTele=$rs2->fetchAssoc();  if($dataChannelTele)  {  $data3["chat\_id"] = $dataChannelTele["chat\_id"];  DB::Insert("outbox\_telegram", $data3 );  }  $rs3 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 2");  $dataChannelAws=$rs3->fetchAssoc();  if($dataChannelAws)  {  $data3["chat\_id"] = $dataChannelAws["chat\_id"];  DB::Insert("outbox\_mail\_aws", $data3 );  }  $rs4 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 3");  $dataChannelWa=$rs4->fetchAssoc();  if($dataChannelWa)  {  $data3["chat\_id"] = $dataChannelWa["chat\_id"];  DB::Insert("outbox\_whatsapp", $data3 );  }  $rs5 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 4");  $dataChannelLine=$rs5->fetchAssoc();  if($dataChannelLine)  {  $data3["chat\_id"] = $dataChannelLine["chat\_id"];  DB::Insert("outbox\_line", $data3 );  }  }  $i += 1;  }  }  }  }  }  else if ($values["message\_general"] == "Private")  {  $data = array();  $data["group\_id"] = $values[group\_id];  $rs = DB::Select("group\_member", $data );  while( $record = $rs->fetchAssoc() )  {  $mmid = $record["member\_id"];  $gmid = $record["group\_member\_id"];  $agendaType = $values["agenda\_type\_id"];  $rs1 = DB::Query("select member\_id from group\_member\_agenda\_subscriber  where agenda\_type\_id = '$agendaType' AND member\_id = '$mmid'");  $dataAgendaType = $rs1->fetchAssoc();  if($dataAgendaType)  {  $data1 = array();  $data1["group\_member\_id"] = $gmid;  $data1["member\_id"] = $mmid;  $data1["group\_agenda\_id"] = $values['group\_agenda\_id'];  DB::Insert("group\_member\_agenda", $data1 );    $data2 = array();  $data2["id\_group\_agenda"] = $values['group\_agenda\_id'];  $data2["id\_member"] = $mmid;  $data2["message\_content"] = $values["message\_content"];  $data2["attachment"] = $values["attachment"];  $data2["date\_send"] = $values["send\_date"];  DB::Insert("group\_agenda\_private", $data2 );  $data3 = array();  $data3["out\_msg"] = $values["message\_content"];  $data3["type"] = "msg";  $data3["flag"] = 1;  $data3["tgl"] = $values["send\_date"];  $rs2 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 1");  $dataChannelTele=$rs2->fetchAssoc();  if($dataChannelTele)  {  $data3["chat\_id"] = $dataChannelTele["chat\_id"];  DB::Insert("outbox\_telegram", $data3 );  }  $rs3 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 2");  $dataChannelAws=$rs3->fetchAssoc();  if($dataChannelAws)  {  $data3["chat\_id"] = $dataChannelAws["chat\_id"];  DB::Insert("outbox\_mail\_aws", $data3 );  }  $rs4 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 3");  $dataChannelWa=$rs4->fetchAssoc();  if($dataChannelWa)  {  $data3["chat\_id"] = $dataChannelWa["chat\_id"];  DB::Insert("outbox\_whatsapp", $data3 );  }  $rs5 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$mmid' AND channel\_id = 4");  $dataChannelLine=$rs5->fetchAssoc();  if($dataChannelLine)  {  $data3["chat\_id"] = $dataChannelLine["chat\_id"];  DB::Insert("outbox\_line", $data3 );  }  }  }  }  if($values['repeat\_type']!='bydate'){  header("Location: group\_agenda\_list.php");  exit();  } |

Kode Program 2. 2 After Record Added Grup Agenda

Kode program 2.2 merupakan kode program yang terletak di event after record grup agenda. Kode program ini berfungsi untuk melakukan insert ke beberapa tabel seperti outbox, general logs, member agenda, agenda private.

|  |
| --- |
| var ctrlLoopType = Runner.getControl(pageid, 'loop\_id');  ctrlLoopType.on('change', function(e){  if (this.getValue() == '1') {  pageObj.toggleItem("integrated\_edit\_field3", true );  pageObj.toggleItem("integrated\_edit\_field9", true );  pageObj.toggleItem("integrated\_edit\_field10", true );  }else{  pageObj.toggleItem("integrated\_edit\_field3", false );  pageObj.toggleItem("integrated\_edit\_field9", false );  pageObj.toggleItem("integrated\_edit\_field10", false );  }  });  var ctrlRepeat = Runner.getControl(pageid, 'repeat');  ctrlRepeat.on('change', function(e){  if (this.getValue() == '0') {  pageObj.toggleItem("integrated\_edit\_field10", false );  }else{  pageObj.toggleItem("integrated\_edit\_field10", true );  }  });  var ctrlgeneral = Runner.getControl(pageid, 'message\_general');    ctrlgeneral.on('change', function(e){  if (this.getValue() == 'General') {  pageObj.toggleItem("integrated\_edit\_field7", false );  pageObj.toggleItem("integrated\_edit\_field2", true );  pageObj.toggleItem("integrated\_edit\_field3", true );  pageObj.toggleItem("integrated\_edit\_field4", true );  pageObj.toggleItem("integrated\_edit\_field6", true );  pageObj.toggleItem("integrated\_edit\_field8", true );  pageObj.toggleItem("integrated\_edit\_field9", true );  pageObj.toggleItem("integrated\_edit\_field10", true );  }else if (this.getValue() == 'Private'){  pageObj.toggleItem("integrated\_edit\_field7", true );  pageObj.toggleItem("integrated\_edit\_field2", false );  pageObj.toggleItem("integrated\_edit\_field3", false );  pageObj.toggleItem("integrated\_edit\_field4", true );  pageObj.toggleItem("integrated\_edit\_field6", true );  pageObj.toggleItem("integrated\_edit\_field8", false );  pageObj.toggleItem("integrated\_edit\_field9", false );  pageObj.toggleItem("integrated\_edit\_field10", false );  }else  {  pageObj.toggleItem("integrated\_edit\_field7", false );  pageObj.toggleItem("integrated\_edit\_field2", false );  pageObj.toggleItem("integrated\_edit\_field3", false );  pageObj.toggleItem("integrated\_edit\_field8", false );  pageObj.toggleItem("integrated\_edit\_field9", false );  pageObj.toggleItem("integrated\_edit\_field10", false );  } |

Kode Program 2. 3 Javascript Onload Event Grup Agenda

Kode program 2.3 merupakan kode program yang terletak di event javascript onload grup agenda. Kode program ini berfungsi untuk melakukan hide kebeberapa inputan yang tidak dibutuhkan.

### Manajemen Agenda Personal

Agenda personal adalah agenda yang dibuat oleh personal. Agenda personal dibuat sebagai reminder terhadap kebutuhan masing-masing personal yang mendaftar.

|  |
| --- |
| $data = array();  $data["member\_id"] = $\_SESSION["member\_id"];  $rs = DB::Select("personal\_channel", $data );  if($rs->fetchAssoc()){  return true;  }else{  $message = "Silahkan Masukkan Personal Channel Terlebbih Dahulu";  return false;  } |

Kode Program 2. 4 Before Record Added Personal Agenda

Kode program 2.4 adalah kode program yang terletak di event before record added personal agenda. Kode program ini berfungsi untuk melakukan pengecekan terhadap channel dari personal, apakah sudah mendaftarkan channelnya atau belum. Jika belum mendaftarkan channel maka tidak dapat membuat agenda.

|  |
| --- |
| $idm = $values["member\_agenda\_id"];  if($values['loop\_type'] == 2)  {  $data = array();  $keyvalues = array();  $data["loop\_value"] = NOW();  $keyvalues["member\_agenda\_id"] = $idm;  DB::Update("personal\_agenda", $data , $keyvalues);  }  $data1 = array();  $data1["out\_msg"] = $values["message\_content"];  $data1["type"] = "msg";  $data1["flag"] = 1;  $tgl = $values["loop\_value"];  $data1["tgl"] = $tgl;  $repeat = $values["repeat"];  if($repeat == 0)  {  $rs1 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$values[member\_id]' AND channel\_id = 1");  $dataChannelTele=$rs1->fetchAssoc();  if($dataChannelTele)  {  $data1["chat\_id"] = $dataChannelTele["chat\_id"];  DB::Insert("outbox\_telegram", $data1 );  }  $rs2 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$values[member\_id]' AND channel\_id = 2");  $dataChannelAws=$rs2->fetchAssoc();  if($dataChannelAws)  {  $data1["chat\_id"] = $dataChannelAws["chat\_id"];  DB::Insert("outbox\_mail\_aws", $data1 );  }  $rs3 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$values[member\_id]' AND channel\_id = 3");  $dataChannelWa=$rs3->fetchAssoc();  if($dataChannelWa)  {  $data1["chat\_id"] = $dataChannelWa["chat\_id"];  DB::Insert("outbox\_whatsapp", $data1 );  }  $rs4 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$values[member\_id]' AND channel\_id = 4");  $dataChannelLine=$rs4->fetchAssoc();  if($dataChannelLine)  {  $data1["chat\_id"] = $dataChannelLine["chat\_id"];  DB::Insert("outbox\_line", $data1 );  }  }  else  {  $i = 0;  $data2 = array();  //$data1["chat\_id"] = $dataChannelTele["chat\_id"];  $id = $values["member\_id"];  $data2["out\_msg"] = $values["message\_content"];  $data2["type"] = "msg";  $data2["flag"] = 1;  $tgl = $values["loop\_value"];  $data2["tgl"] = $tgl;  while($i <= $repeat)  {  $rep = $values["repeat\_type"];  if($rep == "bydate")  {  if($i == 0)  {  $data1["tgl"] = $tgl;  $rs1 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$values[member\_id]' AND channel\_id = 1");  $dataChannelTele=$rs1->fetchAssoc();  if($dataChannelTele)  {  $data1["chat\_id"] = $dataChannelTele["chat\_id"];  DB::Insert("outbox\_telegram", $data1 );  }  $rs2 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$values[member\_id]' AND channel\_id = 2");  $dataChannelAws=$rs2->fetchAssoc();  if($dataChannelAws)  {  $data1["chat\_id"] = $dataChannelAws["chat\_id"];  DB::Insert("outbox\_mail\_aws", $data1 );  }  $rs3 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$values[member\_id]' AND channel\_id = 3");  $dataChannelWa=$rs3->fetchAssoc();  if($dataChannelWa)  {  $data1["chat\_id"] = $dataChannelWa["chat\_id"];  DB::Insert("outbox\_whatsapp", $data1 );  }  $rs4 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$values[member\_id]' AND channel\_id = 4");  $dataChannelLine=$rs4->fetchAssoc();  if($dataChannelLine)  {  $data1["chat\_id"] = $dataChannelLine["chat\_id"];  DB::Insert("outbox\_line", $data1 );  }  }  else  {  }  }  else  {  if($i == 0)  {  $data2["tgl"] = $tgl;  }else if($i > 0)  {    if($values["repeat\_type"] == "daily")  {  $tgl2 = $data2["tgl"];  $tgl3 = date('Y-m-d H:i:s', strtotime($tgl2 . ' +1 day' ));  $data2["tgl"] = $tgl3;  }else if($rep == "weekly")  {  $tgl2 = $data2["tgl"];  $tgl3 = date('Y-m-d H:i:s', strtotime($tgl2 . ' +1 week' ));  $data2["tgl"] = $tgl3;  }else if($rep == "monthly")  {  $tgl2 = $data2["tgl"];  $tgl3 = date('Y-m-d H:i:s', strtotime($tgl2 . ' +1 month' ));  $data2["tgl"] = $tgl3;  }else if($rep == "yearly")  {  $tgl2 = $data2["tgl"];  $tgl3 = date('Y-m-d H:i:s', strtotime($tgl2 . ' +1 year' ));  $data2["tgl"] = $tgl3;  }  }  $rs1 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$id' AND channel\_id = 1");  $dataChannelTele=$rs1->fetchAssoc();  if($dataChannelTele)  {  $data2["chat\_id"] = $dataChannelTele["chat\_id"];  DB::Insert("outbox\_telegram", $data2 );  }  $rs2 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$id' AND channel\_id = 2");  $dataChannelAws=$rs2->fetchAssoc();  if($dataChannelAws)  {  $data2["chat\_id"] = $dataChannelAws["chat\_id"];  DB::Insert("outbox\_mail\_aws", $data2 );  }  $rs3 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$id' AND channel\_id = 3");  $dataChannelWa=$rs3->fetchAssoc();  if($dataChannelWa)  {  $data2["chat\_id"] = $dataChannelWa["chat\_id"];  DB::Insert("outbox\_whatsapp", $data2 );  }  $rs4 = DB::Query("select chat\_id from personal\_channel  where member\_id = '$id' AND channel\_id = 4");  $dataChannelLine=$rs4->fetchAssoc();  if($dataChannelLine)  {  $data2["chat\_id"] = $dataChannelLine["chat\_id"];  DB::Insert("outbox\_line", $data2 );  }  }  $i += 1;  }  }    if($values['repeat\_type']!='bydate'){  header("Location: personal\_agenda\_list.php");  exit();  } |

Kode Program 2. 5 After Record Added Grup Agenda

Kode program 2.5 merupakan kode program yang terletak di event after record added grup agenda. Kode program ini berfungsi untuk melakukan insert data pada tabel outbox sesuai dengan channel yang sudah didaftarkan masing-masing personal.

|  |
| --- |
| var ctrlRepeat = Runner.getControl(pageid, 'repeat');  var ctrlLoopType = Runner.getControl(pageid, 'loop\_type');    ctrlRepeat.on('change', function(e){  if (this.getValue() == '0') {  pageObj.toggleItem("integrated\_edit\_field6", false );  }else{  pageObj.toggleItem("integrated\_edit\_field6", true );  }  });  ctrlLoopType.on('change', function(e){  if (this.getValue() == '1') {  pageObj.toggleItem("integrated\_edit\_field2", true );  pageObj.toggleItem("integrated\_edit\_field5", true );  pageObj.toggleItem("integrated\_edit\_field6", true );  }else{  pageObj.toggleItem("integrated\_edit\_field2", false );  pageObj.toggleItem("integrated\_edit\_field5", false );  pageObj.toggleItem("integrated\_edit\_field6", false );  }  });  this.on('beforeSave', function(formObj, fieldControlsArr, pageObj){  var val = $("#test").val();  formObj.baseParams['test'] = val;  }); |

Kode Program 2. 6 Javascript Onload Personal Agenda

Kode program 2.6 merupakan kode program yang terdapat pada event javascript onload personal agenda. Kode ini berfungsi untuk menyembunyikan inputan yang tidak dibutuhkan.

### Manajemen Grup

#### Group

|  |
| --- |
| $data = array();  $data["member\_id"] = $\_SESSION["member\_id"];  $data["group\_id"] = $values["group\_id"];  $data["user\_type"] = 2;  $data["membership\_type"] = $values['membership\_type'];  $data["valid"] = 'valid';  DB::Insert("group\_member", $data ); |

Kode Program 2. 7 After Record Added Grup

Kode program 2.7 adalah kode program yang terletak di event after record added di tabel group. Kode Program ini berfungsi untuk melakukan insert data pada data yang baru ditambahkan dengan value seperti member\_id, user\_type, dan valid.

|  |
| --- |
| var ctrlTokenType = Runner.getControl(pageid, 'token\_type');    ctrlTokenType.on('change', function(e){  if (this.getValue() == 'token') {  pageObj.toggleItem("integrated\_edit\_field4", true );  }else{  pageObj.toggleItem("integrated\_edit\_field4", false );  }  });  // Place event code here.  // Use "Add Action" button to add code snippets. |

Kode Program 2. 8 JavaScript OnLoad Event Add Page Group

Kode program 2.8 adalah kode program yang terletak di JavaScript OnLoad Event Add Page Group. Kode program ini berfungsi untuk melakukan hide dan show field token pada form add page group. Jika value dari field token type berupa token, maka field token akan muncul, begitu juga sebaliknya.

#### New Group Request

|  |
| --- |
| $data = array();  $data["member\_id"] = $oldvalues['member\_id'];  $rs = DB::Select("personal", $data );  while( $record = $rs->fetchAssoc() )  {  $username = $record["mail"];  }  $data = array();  $data["UserName"] = $username;  $data["GroupID"] = 2;  DB::Insert("project5\_ugmembers", $data );  // Place event code here.  // Use "Add Action" button to add code snippets.  return true; |

Kode Program 2. 9 Before Record Updated New Group Request

Kode program 2.9 adalah kode program yang terletak di event Before Record Updated New Group Request. Kode program ini berfungsi untuk melakukan insert ke tabel project5\_ugmembers berupa username dan GroupID dengan value 2 yang mana memiliki role admin group.

#### Group Product

|  |
| --- |
| if($values['stock\_type']==3){  $data = array();  $data["group\_id"] = $values['group\_id'];  $data["product\_name"] = $values['product\_name'];  $rs = DB::Select("lookup\_stok", $data );  if($rs->fetchAssoc()){  }else{  $message = "Produk belum Ditambahkan di Lookup Stok, tambahkan terlebih dahulu atau ubah stock type";  return false;  }  }  // Place event code here.  // Use "Add Action" button to add code snippets.  return true; |

Kode Program 2. 10 Before Record Added Group Product

Kode program 2.10 adalah kode program yang terletak di event Before Record Added Group Product. Kode program ini berfungsi untuk melakukan validasi ketika menambah product dengan tipe stok API. Jika produk belum ada di lookup stok akan ditampilkan message error dan return false.

|  |
| --- |
| var ctrlStockType = Runner.getControl(pageid, 'stock\_type');    ctrlStockType.on('change', function(e){  if (this.getValue() == '1') {  pageObj.toggleItem("integrated\_edit\_field4", true );  }else{  pageObj.toggleItem("integrated\_edit\_field4", false );  }  });  // Place event code here.  // Use "Add Action" button to add code snippets. |

Kode Program 2. 11 JavaScript OnLoad Event Add Page Group Product

Kode program 2.11 adalah kode program yang terletak di event JavaScript OnLoad Event Add Page Group Product. Kode program ini berfungsi untuk melakukan hide dan show fied stok. Ketika tipe stok yang dimasukkan user defined field token akan ditampilkan jika tidak maka akan di-hide.

#### Group Member

|  |
| --- |
| if(!empty($\_SESSION["group\_id"])){  $pageObject->hideItem("grid\_edit");  }  // Place event code here.  // Use "Add Action" button to add code snippets. |

Kode Program 2. 12 Before Display List Page Group Member

Kode program 2.12 adalah kode program yang terletak di event Before Display List Page Group Member. Kode program ini berfungsi untuk menyembunyikan tombol edit pada list page goup member ketka yang sedang login adalah user biasa.

|  |
| --- |
| $data = array();  $data["group\_id"] = $values[group\_id];  $rs = DB::Select("group", $data );  while( $record = $rs->fetchAssoc() )  {  if($record["token\_type"] == "token"){  $valid = 0;  }else if($record["token\_type"] == "lookup" || $record["token\_type"] == "API"){  $valid = 0;  }else{  $valid = 1;  }  }  $data = array();  $keyvalues = array();  $data["user\_type"] = 3;  $data["membership\_type"] = 3;  $data["valid"] = $valid;  $keyvalues["group\_member\_id"] = $values["group\_member\_id"];  DB::Update("group\_member", $data , $keyvalues); |

Kode Program 2. 13 After Record Added Group Member

Kode program 2.13 adalah kode program yang terletak di event After Record Added Group Member. Kode program ini berfungsi untuk menentukan status validasi dari member group. Jika group bersifat public maka status validasi akan langsung valid, sedangkan jika tidak public maka status validasi akan tidak valid.

|  |
| --- |
| $data = array();  $data["group\_id"] = $oldvalues[group\_id];  $rs = DB::Select("group", $data );  while( $record = $rs->fetchAssoc() )  {  if($record["token\_type"] == "token"){  if($record["token\_group"] == $values['token\_group']){  $data = array();  $keyvalues = array();  $data["valid"] = 1;  $keyvalues["group\_member\_id"] = $oldvalues['group\_member\_id'];  DB::Update("group\_member", $data , $keyvalues);  }  }else if($record["token\_type"] == "lookup" || $record["token\_type"] == "API"){  $data = array();  $data["group\_identifier"] = $values['identifier\_group'];  $data["token"] = $values['token\_personal'];  $res = DB::Select("lookup\_token", $data );  while( $record = $res->fetchAssoc() )  {  $data = array();  $keyvalues = array();  $data["valid"] = 1;  $keyvalues["group\_member\_id"] = $oldvalues['group\_member\_id'];  DB::Update("group\_member", $data , $keyvalues);  }  }  }  // Place event code here.  // Use "Add Action" button to add code snippets. |

Kode Program 2. 14 After Record Updated Group Member

Kode program 2.14 adalah kode program yang terletak di event After Record Updated Group Member. Kode program ini berfungsi untuk menentukan apakah user memasukkan token yang benar apa tidak. Jika token yang dimasukkan user sesuai dengan token yang ada di grup maka member akan otomatis tervalidasi.

|  |
| --- |
| $data = array();  $data["group\_id"] = $values[group\_id];  $rs = DB::Select("group", $data );  while( $record = $rs->fetchAssoc() )  {  if($record["token\_type"] == "token"){  $pageObject->hideItem("integrated\_edit\_field4");  $pageObject->hideItem("integrated\_edit\_field5");  $pageObject->showItem("integrated\_edit\_field3");  }else if($record["token\_type"] == "lookup" || $record["token\_type"] == "API"){  $pageObject->hideItem("integrated\_edit\_field3");  $pageObject->showItem("integrated\_edit\_field4");  $pageObject->showItem("integrated\_edit\_field5");  }  }  if($values['valid'] == 1){  $pageObject->hideItem("integrated\_edit\_field3");  $pageObject->hideItem("integrated\_edit\_field4");  $pageObject->hideItem("integrated\_edit\_field5");  }  // Place event code here.  // Use "Add Action" button to add code snippets. |

Kode Program 2. 15 Before Display Edit Page Group Member

Kode program 2.15 adalah kode program yang terletak di event Before Display Edit Page Group Member. Kode program ini berfungsi untuk menampilkan dan menyembunyikan beberapa field pada edit page group member untuk memasukkan token group atau token pribadi.

### Manajemen Transaksi

#### Group Member Order

|  |
| --- |
| $member\_id = $\_SESSION["member\_id"];  $data = array();  $data["member\_id"] = $\_SESSION["member\_id"];  $data["group\_id"] = $values["group\_id"];  $rs = DB::Select("group\_member", $data );  while( $record = $rs->fetchAssoc() )  {  $group\_member\_id = $record["group\_member\_id"];  }  $data = array();  $keyvalues = array();  $data["member\_id"] = $\_SESSION["member\_id"];  $data["group\_member\_id"] = $group\_member\_id;  $data["valid"] = 'not valid';  $data["payment\_status"] = "not\_paid";  $data["order\_date"] = NOW();  $keyvalues["group\_member\_order"] = $values["group\_member\_order"];  DB::Update("group\_member\_order", $data , $keyvalues); |

Kode Program 2. 16 After Record Added Group Member Order

Kode program 2.16 adalah kode program yang terletak di event After Record Added Group Member Order. Kode Program ini berfungsi untuk melakukan update data pada pesana yang baru saja diinputkan user. Bebrapa field yang diupdate adalah member\_id, satus validasi, status\_payment, dan order\_date.

#### Group Member Order Detail

|  |
| --- |
| $data = array();  $data["group\_member\_order"] = $deleted\_values["group\_member\_order\_id"];  $rs = DB::Select("group\_member\_order", $data );  while( $record = $rs->fetchAssoc() )  {  $payment\_status = $record["payment\_status"];  }  if($payment\_status=="paid"){  $message = "Pesanan yang sudah dibayar tidak dapat dibatalkan";  return false;  }  // Place event code here.  // Use "Add Action" button to add code snippets.  return true; |

Kode Program 2. 17 Before Record Deleted Group Member Order Detail

Kode program 2.17 adalah kode program yang terletak di event Before Record Deleted Group Member Order Detail. Kode program ini berfungsi untuk untuk melakukan validasi ketika member menghapus detail pesanan. Ketika pesanan sudah dibayarkan, pengguna tidak dapat menghapus pesanan tersebut.

|  |
| --- |
| $data = array();  $data["group\_member\_order"] = $values['group\_member\_order\_id'];  $rs = DB::Select("group\_member\_order", $data );  while( $record = $rs->fetchAssoc() )  {  $group\_id = $record["group\_id"];  }  $data = array();  $data["group\_product\_id"] = $values['group\_product\_id'];  $rs = DB::Select("group\_product", $data );  while( $record = $rs->fetchAssoc() )  {  $stokType = $record["stock\_type"];  $stok = $record["stock"];  $namaProduk = $record["product\_name"];  }  if($stokType==1){  if($stok<$values['quantity']){  $message = "Stok tinggal ".$stok;  return false;  }  }else if($stokType==3){  $data = array();  $data["group\_id"] = $group\_id;  $data["product\_name"] = $namaProduk;  $rs = DB::Select("lookup\_stok", $data );  while( $record = $rs->fetchAssoc() )  {  $stokAPI = $record["stock"];  }    if($stokAPI<$values['quantity']){  $message = "Stok tinggal ".$stokAPI;  return false;  }  return true;  }  // Place event code here.  // Use "Add Action" button to add code snippets.  return true; |

Kode Program 2. 18 Before Record Added Group Member Order Detail

Kode program 2.18 adalah kode program yang terletak di event Before Record Added Group Member Order Detail. Kode program ini berfungsi untuk melakukan validasi ketika menambahkan detail pesanan. Ketika jumlah pesanan melebihi stok yag ada, maka akan muncul pesan yang menginformasikan stok yang tersedia.

|  |
| --- |
| $member\_id = $\_SESSION["member\_id"];  $data = array();  $data["group\_member\_order"] = $values['group\_member\_order\_id'];  $rs = DB::Select("group\_member\_order", $data );  while( $record = $rs->fetchAssoc() )  {  $group\_id = $record["group\_id"];  $total = $record["total"];  }  $data = array();  $data["member\_id"] = $\_SESSION["member\_id"];  $data["group\_id"] = $group\_id;  $rs = DB::Select("group\_member", $data );  while( $record = $rs->fetchAssoc() )  {  $group\_member\_id = $record["group\_member\_id"];  }  $data = array();  $data["group\_product\_id"] = $values["group\_product\_id"];  $rs = DB::Select("group\_product", $data );  while( $record = $rs->fetchAssoc() )  {  $nominal = $record["price"];  }  $total\_detail = $nominal\*$values["quantity"];  $data = array();  $keyvalues = array();  $data["group\_member\_id"] = $group\_member\_id;  $data["member\_id"] = $\_SESSION["member\_id"];  $data["group\_id"] = $group\_id;  $data["nominal"] = $nominal;  $data["total"] = $total\_detail;  $data["progress"] = "order";  $keyvalues["group\_member\_order\_detail\_id"] = $values["group\_member\_order\_detail\_id"];  DB::Update("group\_member\_order\_detail", $data, $keyvalues );  $total\_update = $total + $total\_detail;  $data = array();  $keyvalues = array();  $data["total"] = $total\_update;  $keyvalues["group\_member\_order"] = $values["group\_member\_order\_id"];  DB::Update("group\_member\_order", $data , $keyvalues); |

Kode Program 2. 19 After Record Added Group Member Order Detail

Kode program 2.19 adalah kode program yang terletak di event After Record Added Group Member Order Detail. Kode Program ini berfungsi untuk melakukan update pada table group\_member\_order\_detail berupa nominal, total, dan progress. Selain itu juga dilakukan update pada tabel group\_member\_order berupa total pesanan.

#### Group Member Checkin

|  |
| --- |
| $sql = "SELECT \* FROM group\_member\_checkin WHERE (checkin\_date > '" . $values['checkin\_date'] .  "' AND checkin\_date < '" . $values['checkout\_date'] .  "' ) OR ( checkin\_date < '" . $values['checkin\_date'] .  "' AND checkout\_date > '" . $values['checkin\_date'] . "' )";  $preparedSQL = DB::PrepareSQL( $sql );  $result = DB::Query( $preparedSQL );  if($result->fetchAssoc()) {  $message = "Room/table sudah tidak tersedia pada waktu yang anda pilih";  return false;  }  // Place event code here.  // Use "Add Action" button to add code snippets.  return true; |

Kode Program 2. 20 Before Record Added Group Member Checkin

Kode program 2.20 adalah kode program yang terletak di event Before Record Added Group Member Checkin. Kode Program ini berfungsi untuk melakukan validasi terhadap room yang akan dicheckin, jika sudah terdapat checkin room yang sama di tanggal yang sama maka akn menampikan pesan.

|  |
| --- |
| $data = array();  $data["member\_id"] = $values[member\_id];  $data["group\_id"] = $values[group\_id];  $rs = DB::Select("group\_member", $data );  while( $record = $rs->fetchAssoc() )  {  $group\_member\_id = $record["group\_member\_id"];  }  $data = array();  $keyvalues = array();  $data["group\_member\_id"] = $group\_member\_id;  $data["valid"] = 'not valid';  $keyvalues["group\_member\_checkin\_id"] = $values[group\_member\_checkin\_id];  DB::Update("group\_member\_checkin", $data , $keyvalues); |

Kode Program 2. 21 After Record Added Group Member Checkin

Kode program 2.21 adalah kode program yang terletak di event After Record Added Group Member Checkin. Kode program ini berfungsi untuk melakukan update pada tabel group\_member\_checkin dengan value berupa group\_member\_id dan status validasi.

#### Member Checkin

|  |
| --- |
| $data = array();  $data["member\_id"] = $\_SESSION["member\_id"];  $data["group\_id"] = $values[group\_id];  $rs = DB::Select("group\_member", $data );  while( $record = $rs->fetchAssoc() )  {  $group\_member\_id = $record["group\_member\_id"];  }  $data = array();  $keyvalues = array();  $data["group\_member\_id"] = $group\_member\_id;  $data["valid"] = 'not valid';  $data["member\_id"] = $\_SESSION["member\_id"];  $keyvalues["group\_member\_checkin\_id"] = $values[group\_member\_checkin\_id];  DB::Update("group\_member\_checkin", $data , $keyvalues); |

Kode Program 2. 22 After Record Added Member Checkin

Kode program 2.22 adalah kode program yang terletak di event After Record Added Member Checkin. Kode program ini berfungsi untuk melakukan update pada tabel group\_member\_checkin dengan value berupa group\_member\_id, member\_id dan status validasi.

#### Validate Order

|  |
| --- |
| $data = array();  $data["group\_id"] = $oldvalues["group\_id"];  $rs = DB::Select("group", $data );  while( $record = $rs->fetchAssoc() )  {  $group = $record["group\_name"];  }  $data = array();  $data["group\_member\_order\_id"] = $oldvalues["group\_member\_order"];  $rs = DB::Select("group\_member\_order\_detail", $data );  while( $record = $rs->fetchAssoc() )  {  $group = $record["group\_name"];    $data = array();  $keyvalues = array();  $data["progress"] = "on\_progress";  $keyvalues["group\_member\_order\_detail\_id"] = $record["group\_member\_order\_detail\_id"];  DB::Update("group\_member\_order\_detail", $data , $keyvalues);  } |

Kode Program 2. 23 After Record Updated Validate Order

Kode program 2.23 adalah kode program yang terletak di event After Record Updated Validate Order. Kode program ini berfungsi untuk melakukan update pada tabel group\_member\_order\_detail berupa progress dengan value on\_progress.

#### Validate Checkin

|  |
| --- |
| group\_id = ':session.group\_id' and valid = 'not valid' |

Kode Program 2. 24 Additional Where Clause Validate Checkin

Kode program 2.24 adalah kode program yang terletak di additional where clause pada validate checkin. Kode program ini berfungsi untuk memfilter data yang ditampilkan hanya milik group tertentu dan status not\_valid.

#### Update Order Progress

|  |
| --- |
| group\_member\_order\_detail.group\_id = ':session.group\_id' and valid = 'valid' and group\_member\_order\_detail.progress = "on\_progress" |

Kode Program 2. 25 Additional Where Clause Update Order Progress

Kode program 2.25 adalah kode program yang terletak di additional where clause pada update order progress. Kode program ini berfungsi untuk memfilter data yang ditampilkan hanya milik group tertentu dan status valid, dan progressnya on\_progress.

#### Payment

|  |
| --- |
| $data = array();  $data["group\_member\_order\_id"] = $oldvalues["group\_member\_order"];  $rs = DB::Select("group\_member\_order\_detail", $data );  while( $record = $rs->fetchAssoc() )  {  $data = array();  $data["group\_product\_id"] = $record['group\_product\_id'];  $rs = DB::Select("group\_product", $data );  while( $record2 = $rs->fetchAssoc() )  {  $stokType = $record2["stock\_type"];  $stok = $record2["stock"];  $namaProduk = $record2["product\_name"];  }  if($stokType==1){  $data = array();  $keyvalues = array();  $data["stock"] = $stok - $record['quantity'];  $keyvalues["group\_product\_id"] = $record['group\_product\_id'];  DB::Update("group\_product", $data , $keyvalues);  }else if($stokType==3){  $data = array();  $data["group\_id"] = $record['group\_id'];  $data["product\_name"] = $namaProduk;  $rs = DB::Select("lookup\_stok", $data );  while( $record3 = $rs->fetchAssoc() )  {  $stokAPI = $record3["stock"];  }    $data = array();  $keyvalues = array();  $data["stock"] = $stokAPI - $record['quantity'];  $keyvalues["group\_id"] = $record['group\_id'];  $keyvalues["product\_name"] = $namaProduk;  DB::Update("lookup\_stok", $data , $keyvalues);  }  }  // Place event code here.  // Use "Add Action" button to add code snippets.  return true; |

Kode Program 2. 26 Before Record Updated Payment

Kode program 2.26 adalah kode program yang terletak di event Before Record Updated Payment. Kode program ini berfungsi untuk melakuka pengurangan stok produk. Stok produk yang ada di tabel group\_product maupun lookup\_stok akan dikurangin sesuai dengan jumlah pesanan keita pembayaran telah dilakukan.

|  |
| --- |
| $data = array();  $data["group\_id"] = $oldvalues["group\_id"];  $rs = DB::Select("group", $data );  while( $record = $rs->fetchAssoc() )  {  $group = $record["group\_name"];  }  $data = array();  $keyvalues = array();  $data["payment\_status"] = "paid";  $data["change\_money"] = $values["money\_received"] - $oldvalues["total"];  $keyvalues["group\_member\_order"] = $oldvalues["group\_member\_order"];  DB::Update("group\_member\_order", $data , $keyvalues); |

Kode Program 2. 27 After Record Updated Payment

Kode program 2.27 adalah kode program yang terletak di event After Record Updated Payment. Kode program ini berfungsi untuk update tabel group\_member\_order berupa payment statusnya paid dan nilai uang kembalian.

#### Review Checkin

|  |
| --- |
| valid = 'valid' and member\_id = ':session.member\_id' and rating\_member is null |

Kode Program 2. 28 Additional Where Clause Review Checkin

Kode program 2.28 adalah kode program yang terletak di additional where clause review checkin. Kode program ini berfungsi untuk memfilter data yang ditampilkan hanya milik member tertentu, ratingnya null dan status valid.

#### Review Order

|  |
| --- |
| valid = 1 and member\_id = ':session.member\_id' and rating\_member is null |

Kode Program 2. 29 Additional Where Clause Review Order

Kode program 2.29 adalah kode program yang terletak di additional where clause review order. Kode program ini berfungsi untuk memfilter data yang ditampilkan hanya milik member tertentu, ratingnya null dan status valid.

## Panduan Pengoperasian Sistem

Sistem ini memiliki tiga role yaitu admin puncak, admin grup, dan user. Admin puncak merupakan admin yang memiliki sistem ini, admin grup adalah admin yang memiliki suatu grup, dan user daladh pengguna biasa. Keterangan lebih lanjut dari masing-masing role dapat dilihat pada tabel berikut.

Tabel 2. 1 Daftar Role

|  |  |
| --- | --- |
| **Nama Role** | **Keterangan** |
| Admin Puncak | Admin puncak merupakan admin yang memiliki sistem. Admin puncak dapat melakukan berbagai hal diantaranya CRUD *Order Type*, *Stock Type*, *Product Type*, *Province*, *Channel*, *Country*, *Loop Type*, *Bussiness Type*, *Membership Type*, *Group User Type*, dan menerima permintaan pembuatn grup baru pada menu *New Group Request*. |
| Admin Group | Admin grup merupakan admin dari suatu grup yang terdapat pada sistem. Admin grup dapat melakukan berbagia hal pada sistem ini, diantaranya membuat grup baru, menambahkan grup agenda type, mebambahkan agenda grup general, menambahkan grup agenda private, menambahkan produk grup, mebambahkan token grup dan token personal, melakukan validasi chekin, melkaukan pembarayan pesanan, melakukan validasi pesanan, dan melakukan update progress pesanan. |
| User | User merupakan prngguna biasa pada sistem ini. User biasa pada sistem ini dapat melakukan berbagai hal diantaranya mendaftarkan akun baru, bergabung dengan grup, membuat personal channel, membuat personal agenda, membuat pesanan tanpa checkin, melakukan checkin, membuat pesanan setelah checkin, memberikan ulasan pesanan, dan memberikan ulasan checkin. |

Tabel 2.1 merupakan daftar dan keterangan dari masing-masing role. Admin puncak merupakan admin yang memiliki sistem ini, admin grup adalah admin yang memiliki suatu grup, dan user daladh pengguna biasa. Keterangan lebih lanjut dari masing-masing role dapat dilihat pada tabel berikut. Adapan daftar akun pada sistem dari masing-masing role adalah sebagai berikut.

Tabel 2. 2 Daftar Akun Setiap Role

|  |  |  |
| --- | --- | --- |
| **Role** | **Email** | **Password** |
| Admin Puncak | hyper@admin.com | 123123 |
| Admin Group | jyotisa1616@gmail.com | 123123 |
| User | nyomanjyotisabali@gmail.com | 123123 |
| User | nyomanjyotisabali@gmail.com | 123123 |

Tabel 2.2 merupakan daftar akun dari setiap role yang da pada sistem. Jika login kesistem menggunakan salah satu dari akun diatas maka akan langsung mendapatkan role sesuai yang tertera pada tabel di atas.

### Panduan Pengoperasian Admin Puncak

Admin puncak dalam sistem ini dapat melakukan berbagai hal diantaranya CRUD *Order Type*, *Stock Type*, *Product Type*, *Province*, *Channel*, *Country*, *Loop Type*, *Bussiness Type*, *Membership Type*, *Group User Type*, dan menerima permintaan pembuatn grup baru pada menu *New Group Request*.

#### Menerima Permintaan Pembuatan Grup Baru

Sebelum suatu grup dapat beroperasi, admin puncak harus melakukan validasi terhadap grup baru tersebut. Admin puncak dapat memilih menu New Group Request untuk melihat daftar permintaan pembuatan grup baru seperti pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 5 Halaman List New Group Request

Gambar 2.5 merupakan tampilan halaman list New Grup Request yang ada di sistem admin puncak. List yang ada disini adalah list group yang baru saja dibuat oleh calon admin group yang berlum divalidasi. Setelah login sebagai admin puncak, admin puncak dapat melakukan validasi terhadap grup yang sebeumnya sudah dibuat user dengan mengklik tombol Approve. Tampilan setelah klik tombol Approve dapat dilihat pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 6 Validasi Group Request

Gambar 2.6 merupakan tampilan dari inline edit pada halaman New Group Request. Pada halaman inline edit ini admin puncak haya dapat mengubah status validasi menjadi valid atau tetap membuarkan statusnya not valid. Setelah group baru divalidasioleh admin maka user yang sebelumnya membuat grup mendapatkan role admin grup dan ketika admin group tersebut login akan mendapatkan dashboard untuk mengelalo groupnya.

### Panduan Pengoperasian Admin Group

Admin grup dapat melakukan berbagia hal pada sistem ini, diantaranya membuat grup baru, menambahkan grup agenda type, mebambahkan agenda grup general, menambahkan grup agenda private, menambahkan produk grup, mebambahkan token grup dan token personal, melakukan validasi chekin, melkaukan pembarayan pesanan, melakukan validasi pesanan, dan melakukan update progress pesanan.

#### Membuat Grup Baru

Pengguna yang yang telah login ke sistem dapat memilih untuk membuat grup apda menu grup da menekan tombol tambah baru untuk membuat grup baru. Berikut merupakan tampilan dari halaman list grup yang ada di sistem.

|  |
| --- |
|  |

Gambar 2. 7 Halaman List Group

Gambar 2.7 merupakan tampilan dari halaman list grup. Pada halaman ini akan tampil semua grup yang ada di sistem yang telah divalidasi oleh admin puncak. Pengguna dengan role user dapat membuat grup dari halaman ini dengan memilih tombol tambah baru dan akan muncul form untuk membuat grup seperti pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 8 Halaman Tambah Grup

Gambar 2.8 merupakan tampilan dari form pembuatan grup. Terdapat Sepuluh inputan yang harus dimasukan oleh user pada form pembuatan grup yaitu Nama Grup, Deskripsi, Logo, Tipe Token, Token, Tipe Keanggotaan, Tipe Bisnis, Waktu Buka, Waktu Tutup, dan Hari Libur. User yang telah memasukkan inputan dapat melakukan pembuatan grup dengan memklik tombol Simpan.

|  |
| --- |
|  |

Gambar 2. 9 Halaman List Your Group

Gambar 2.9 merupakan tampilan halaman list your grup. Grup yang sebelumnya dibuat oleh user akan tampil di halaman ini. Namun grup yang dibuat user sebelumnya masih belum valid sehingga perlu divalidasi oleh admin puncak terlebuh dahulu.

|  |
| --- |
|  |

Gambar 2. 10 Halaman Your Group Setelah Validasi

Gambar 2.10 merupakan tampilan dari halaman list Your Group dari pengguna yang membuat grup sebelumnya. Pada halaman ini group yang sebelumnya sudah dibuat akan memiliki status validasi valid dan dan user ini mendapatkan role admin group sehingga dapat menambahkan product dari grup.

#### Menambahkan Grup Agenda Type

Grup agenda type adalah tipe agenda dari grup agenda. Grup agenda type harus didaftarkan terlebih dahulu sebelum bisa digunakan. Berikut merupakan cara mendaftarkan grup agenda type.

|  |
| --- |
|  |

Gambar 2. 11 Halaman List Grup Agenda Type

Gambar 2.11 merupakan tampilan daftar dari grup agenda type yang sudah didaftarkan. Selanjutnya cara untuk mendaftarkan grup agenda type baru adalah dengan klik tombol add new di pojok kiri atas halaman web.

|  |
| --- |
|  |

Gambar 2. 12 Tampilan Awal Pembuatan Grup Agenda Type

Gambar 2.12 merupakan tampilan untuk mendaftarkan grup agenda type. Terlihat terdapat inputan group untuk memilih grup yang dimiliki oleh admin grup tersebut. Kemudian terdapat inputan agenda type untuk mendaftarkan agenda type baru. Kemudian terdapat deskripsi untuk mengisi deskripsi dari agenda type tersebut. Saat admin klik tombol save maka akan dilakukan request persetujuan untuk membuat agenda type baru kepada admin utama.

#### Menambahkan Agenda Grup General

Agenda grup adalah agenda yang dibuat oleh admin grup untuk disebarkan kepada member grup yang melakukan subscriber kepada agenda type pada grup tersebut. Berikut merupakan tampilan awal dari list agenda yang sudah dibuat pada suatu grup.

|  |
| --- |
|  |

Gambar 2. 13 Tampilan Halaman Daftar Agenda

Gambar 2.13 merupakan tampilan awal atau daftar dari agenda yang sudah dibuat pada grup tersebut. Terlihat terdapat tombol untuk melihat detail agenda, detail dari tabel yang terhubung dengan grup agenda tersebut. Terdapat pula nama grup, tipe agenda, tipe loop, tanggal loop, jenis pesan, konten pesan tanggal kirim (untuk private), lampiran, repeat dan repeat type. Untuk membuat agenda grup general baru yakni dengan mengklik tombol add new.

|  |
| --- |
|  |

Gambar 2. 14 Tampilan Awal Pembuatan Grup Agenda General

Gambar 2.14 merupakan tampilan awal saat melakukan pembuatan agenda grup. Terlihat terdapat inputan grup untuk memilih grup dari grup yang dimiliki admin grup. Kemudian terdapat inputan agenda type yang merupakan tipe agenda grup yang sudah didaftarkan sebelumnnya. Kemudian message general yang merupakan jenis dari agenda grup, terdapat 2 pilihan yakni general untuk mengirim secara umum kepada member yang sudah menjadi subscriber pada suatu agenda type, dan private yang dikirim kepada member secara private.

|  |
| --- |
|  |

Gambar 2. 15 Contoh Pengisian Pembuatan Agenda Grup General Real Time

Gambar 2.15 merupakan tampilan pembuatan agenda grup saat dipilih message general tipe general. Terdapat tambahan beberapa inputan seperti agenda name yang merupakan judul atau nama dari agenda grup tersebut, kemudian message content yakni isi pesan atau konten dari agenda grup tersebut, kemudian terdapat attachment yakni lampiran dari agenda grup yang berjenis file, kemudian inputan loop yang merupakan pilihan pengiriman, terdapat 2 pilihan yakni real time untuk mengirim agenda grup tersebut pada tanggal tersebut, atau on date yakni mengirim agenda grup berdasarkan inputan tanggal yang diisi oleh admin yang akan dijelaskan selanjutnya. Jika admin grup memilih loop real time maka saat mengklik tombol save akan terbentuk agenda grup general yang dikirim kepada member yang sudah melakukan subscriber ada agenda type tersebut pada tanggal saat itu.

|  |
| --- |
|  |

Gambar 2. 16 Contoh Pengisian Pembuatan Agenda Grup General Ondate

Gambar 2.16 merupakan tampilan saat admin grup memilih loop on date. Terlihat saat memilih on date maka akan tampil inputan tambahan berupa loop value yang merupakan tanggal pengiriman agenda grup, kemudia terdapat juga repeat yakni jumlah pengulangan pengiriman agenda yang diinginkan, jika diisi 0 maka tidak melakukan perulangan dan ketika di click tombol save maka akan membuat agenda grup pada tanggal berdasarkan loop value tersebut. Namun jika diisi dengan angka lebih dari 0 maka akan terdapat inputan tambahan yang akan dijelaskan selanjutnya.

|  |
| --- |
|  |

Gambar 2. 17 Contoh Pengisian Pembuatan Agenda Grup General Bydate

Gambar 2.17 merupakan tampilan saat mengisi nilai repeat dengan angka lebih dari 0. Terlihat terdapat tambahan repeat type yang memiliki beberapa pilihan yakni daily untuk melakukan perulang ditambah 1 hari dari tanggal loop value sebanyak nilai inputan repeat. Kemudian terdapat pilihan weekly untuk melakukan perulang ditambah 1 minggu dari tanggal loop value sebanyak nilai inputan repeat. Kemudian terdapat pilihan monthly untuk melakukan perulang ditambah 1 bulan dari tanggal loop value sebanyak nilai inputan repeat. Kemudian terdapat pilihan yearly untuk melakukan perulang ditambah 1 tahun dari tanggal loop value sebanyak nilai inputan repeat. Yang mana jika memilih pilihan repeat type dari salah satu pilihan tersebut, saat admin grup meng-click tombol save maka akan dibuat agenda perulangan sebanyak inputan yang telah diisi tersebut. Terakhir terdapat pilihan bydate yang akan dijelaskan selanjutnya.

|  |
| --- |
|  |

Gambar 2. 18 Tampilan Inputan Tanggal Dari Repeat Type Bydate

Gambar 2.18 merupakan tampilan saat admin grup memilih repeate type bydate pada pembuatan grup agenda. Terlihat terdapat nama dari grup agenda yakni pemberitahuan yang telah dibuat sebelumnya, kemudian terdapat inputan dodate yang berupa tanggal yang ingin dilakukan perulangan grup agenda. Jumlah inputan dodate tergantung dari nilai inputan repeat sebelumnya. Saat admin grup meng-click save maka akan dibuat grup agenda dengan perulangan sebanyak nilai repeat, serta tanggal sesuai dengan dodate.

|  |
| --- |
|  |

Gambar 2. 19 Tampilan Daftar Agenda Setelah Selesai

Gambar 2.19 merupakan tampilan setelah selesai membuat agenda grup dengan type repeat date bydate. Terlihat terdapat daftar agenda yang melakukan repeat bydate.

|  |
| --- |
|  |

Gambar 2. 20 Daftar Agenda Grup Dengan Daftar Agenda General Log

Gambar 2.20 merupakan tampilan daftar agenda grup dengan daftar agenda general lognya. Terlihat terdapat agenda grup dengan daftar general lognya.

|  |
| --- |
|  |

Gambar 2. 21 Daftar Agenda Grup Dengan Daftar Member Agenda

Gambar 2.21 merupakan tampilan daftar agenda grup dengan daftar member agenda. Terlihat terdapat daftar agenda grup beserta daftar agenda dari member yang melakukan subscriber terhadap jenis agendanya.

|  |
| --- |
|  |

Gambar 2. 22 Daftar Agenda Grup Dengan Daftar Repeat Agenda

Gambar 2.22 merupakan tampilan daftar agenda grup dengan daftar repeat agenda. Terlihat terdapat daftar agenda beserta daftar repeatnya yang bedasarkan tanggal. Hal ini berlaku hanya jika memilih repeat type bydate.

#### Menambahkan Grup Agenda Private

Agenda grup private adalah agenda yang dibuat oleh admin grup untuk disebarkan kepada member grup secara private. Berikut merupakan tampilan awal dari list agenda yang sudah dibuat pada suatu grup.

|  |
| --- |
|  |

Gambar 2. 23 Tampilan Daftar Grup Agenda Private

Gambar 2.23 merupakan tampilan awal atau daftar dari agenda yang sudah dibuat pada grup tersebut. Terlihat terdapat tombol untuk melihat detail agenda, detail dari tabel yang terhubung dengan grup agenda tersebut. Terdapat pula nama grup, tipe agenda, tipe loop, tanggal loop, jenis pesan, konten pesan tanggal kirim (untuk private), lampiran, repeat dan repeat type. Untuk membuat agenda grup private baru yakni dengan mengklik tombol add new.

|  |
| --- |
|  |

Gambar 2. 24 Tampilan Awal Pembuatan Grup Agenda Private

Gambar 2.24 merupakan tampilan awal saat melakukan pembuatan agenda grup private. Terlihat terdapat inputan message general yang merupakan jenis dari agenda grup, terdapat 2 pilihan yakni general untuk mengirim secara umum kepada member yang sudah menjadi subscriber pada suatu agenda type, dan private yang dikirim kepada member secara private. Apabila dipilih private maka hanya akan muncul field Agenda Name dan Send Date.

|  |
| --- |
|  |

Gambar 2. 25 Contoh Pengisian Pembuatan Agenda Grup Private

Gambar 2.25 merupakan tampilan pembuatan agenda grup saat dipilih message general tipe private. Terdapat tambahan beberapa inputan seperti agenda name yang merupakan judul atau nama dari agenda grup private tersebut, kemudian send date yang merupakan tanggal pengiriman agenda grup private tersebut. Kemudian saat klik save maka akan dibuat grup agenda private dan akan langsung diarahkan ke menu import Group Agenda Private.

|  |
| --- |
|  |

Gambar 2. 26 Tampilan Memilih File Import Grup Agenda Private

Gambar 2.26 merupakan tampilan setelah mengklik import pada tahap sebelumnya. Terlihat terdapat tombol browse untuk memilik file excel yang akan diimport. Untuk mendapatkan list anggota group bisa didapat dari export group member dari sisi admin, maka akan didapatkan file excel yang bisa dimodifikasi sesuai keperluan untuk digunakan di fitur import ini.

|  |
| --- |
|  |

Gambar 2. 27 Tampilan Data Import Grup Agenda Private

Gambar 2.27 merupakan tampilan setelah memilih file excel yang ingin di import. Terlihat terdapat beberapa data yang berasal dari excel yang telah dipilihs sebelumnnya. Selanjutnya adalah mengklik tombol import data, maka data tersebut akan masuk ke grup agenda personal yang sudah dibuat sebelumnnya.

|  |
| --- |
|  |

Gambar 2. 28 Pesan Berhasil Melakukan Import Grup Agenda Private

Gambar 2.28 merupakan tampilan saat berhasil melakukan import data dari excel yang telah dipilih sebelumnya. Terlihat terdapat pesan berhasil pada dua data yang sebelumnnya di import.

|  |
| --- |
|  |

Gambar 2. 29 Detail Penerima Grup Agenda Private

Gambar 2.29 merupakan list dari grup agenda private dengan detail member yang menerima pesan private tersebut.

#### Menambahkan Produk Grup

Sebagai seorang admin grup, pengguna dapat melakukan penambahan produk baru dari grup tersebut. Untuk melakukan penambahan produk baru admin grup dapat membuka menu grup product dan menekan tombol tambah baru.

|  |
| --- |
|  |

Gambar 2. 30 Halaman Awal List Group Product

Gambar 2.30 merupakan tampilan halaman Group Product dari admin grup. Pada halaman ini akan muncul list product dari grup dan tombol Tambah Baru untuk menambahkan produk baru. Tampilan dari halaman tambah baru produk dapat dilihat pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 31 Tambah Product Tipe Stok Always Ready dan Tipe Order Member

Gambar 2.31 merupakan tampilan dari halaman tambah produk baru. Terdapat delapan inputan yang harus dimasukan oleh admin grup pada halaman ini yaitu Product Type, Product Name, Stock Type, Stock, Order Type, Price, Currency dan Image. Terdapat tiga jenis Stock Type yaitu defined, always ready, dan API, dan juga terdapat tiga tipe order yaitu public, member, dan member chekcin. Pada gambar diatas merupakan contoh dari tipe stok always ready dan order type member. Admin Grup yang telah memasukkan semua inputan dapat menambahkan produk baru dengan mengklik tombol “Masuk”.

|  |
| --- |
|  |

Gambar 2. 32 Tambah Product Tipe Defined dan Tipe Order Public

Gambar 2.32 merupakan tampilan dari halaman tambah produk baru. Pada gambar diatas merupakan contoh dari menambahkan produk dengan tipe stok defined dan order type public. Admin Grup yang telah memasukkan semua inputan dapat menambahkan produk baru dengan mengklik tombol “Masuk”.

|  |
| --- |
|  |

Gambar 2. 33 Pemberitahuan untuk Menambah Data di Lookup Stok

Gambar 2.33 merupakan tampilan dari halaman tambah produk baru. Pada gambar diatas merupakan contoh dari menambahkan produk dengan tipe API. Namun muncul pesan error dari sistem yang meninformasikan bahwa produk belum ditambahkan di Loopkup Stok. Oleh karena itu, unutk dapat menggunakan stock type API maka harus menambahkan produk ke lookup stok seperti pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 34 Halaman Tambah Baru Lookup Stok

Gambar 2.34 merupakan tampilan dari tambah baru loopup stok. Terdapat dua inputan yang harus dimasukan oleh admin group pada halaman ini yaitu Product Name dan Stock. Setelah menambahkan produk dan stoknya di lookup stok admi group dapat menambahkan produk baru yang menggunakan tipe stok API.

|  |
| --- |
|  |

Gambar 2. 35 Tambah Product Tipe API Ready dan Tipe Order Member Checkin

Gambar 2.35 merupakan tampilan dari halaman tambah produk baru. Pada gambar diatas merupakan contoh dari menambahkan produk dengan tipe stok API dam tipe order checkin. Produk ini hanya akan muncul pada proses pemesanan yang dilakukan oleh member yang telah checkin dan stoknya akan diintegrasikan dengan tabel loopkup stok.

|  |
| --- |
|  |

Gambar 2. 36 Halaman List Product Setelah Iput Data

Gambar 2.36 merupakan tampilan dari halaman list produk. Pada halaman ini muncul semua product yang telah dibuat sebelumnya dengan beberapa detail yang dapat ditampilkan. Pada tiap produk uga terdapat gambar dari produk tersebut yang ditampilkan dengan format thumbnail.

#### Menambahkan Token Grup dan Token Personal

Admin grup dapat memodifikasi tipe token dari grupnya masing masing. Terdapat tiga tipe yaitu public, token dan lookup. Ketiga tipe tersebut memiliki fungsi yang berbeda. Tampilan malukan penambahakn token grup dapat dilihat pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 37 Edit Token Type Menjadi Token

Gambar 2.37 merupakan tampilan dari halaman edit your grup di sisi admin grup. Pada gambar diatas dilakukan perubahan pada token type dari yang sebelumnya public menjadi token dan menambahkan token grup dengan value 185241.

|  |
| --- |
|  |

Gambar 2. 38 Edit Token Type Menjadi Lookup

Gambar 2.38 merupakan tampilan dari halaman edit your grup di sisi admin grup. Pada gambar diatas dilakukan perubahan pada token type dari yang sebelumnya token menjadi lookup. Selanjutnya untuk dapat menggunakan fitur lookup harus dilakukan input data di lookup token.

|  |
| --- |
|  |

Gambar 2. 39 Halaman Tambah Lookup Token

Gambar 2.39 merupakan tampilan dari halaman tambah baru lookup token. Pada gambar diatas penambahan data baru yaitu pada field Group Identifier dan Token. Pada Group Indentifier dimasukkan value NIM dan pada Token dimasukkan value 1905551092.

#### Validasi Checkin

Setelah user melakukan chekcin, admin grup dapat melakukan validasi terhadap checkin yang telah dibuat tersebut. Setalah melakukan validasi, anggota group yang bersangkutan sudah dapat melakukan pemesanan produk dengan tipe checkin member.

|  |
| --- |
|  |

Gambar 2. 40 Halaman List Validasi Checkin

Gambar 2.40 merupakan tampilan dari halaman list validasi checkin dari sisi admin grup. Padahalaman ini admin grup dapat melakukan validasi dengan menekan tombol validate dan akan muncul fitur inline edit seperti pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 41 Inline Edit Validasi Checkin

Gambar 2.41 merupakan tampilan dari inline edit pada halmaan list validasi checkin. Admin grup hanya dapat melakukan perubahan pada field validasi degan mengubah menjadi valid atau tetap not valid.

#### Pembayaran Pesanan

Setelah anggota grup melakukan pemesanan admin grup harus melakukan pembayaran dengan memasukkan uang yang diterima, dan akan muncul uang kembalian yang hasrus diberikan admin grup kepada pengguna.

|  |
| --- |
|  |

Gambar 2. 42 Halaman List Pembayaran

Gambar 2.42 merupakan tampilan dari halaman list pembayaran. Pada halaman ini admin grup dapat melakukan pembayaran dengan mengklik tombol Make Payment maka akan muncul halaman pembayaran yang dapat dilakukan oleh admin grup seperti pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 43 Halaman Tambah Pembayaran

Gambar 2.43 merupakan tampilan dari halaman pembayaran yang dilakukan oleh admin grup. Pada halaman ini akan mucul total dari pesanan dan field uang diterima yang harus dimasukkan oleh admin grup dengan jumlah yang sama atau lebih besar dari total. Setelah klik tombo simpan akan tampil halaman view seperti berikut.

|  |
| --- |
|  |

Gambar 2. 44 Halaman View Uang Kembalian

Gambar 2.44 merupakan tampilan dari halaman view yang menampilkan uang kembalian setelah melakukan proeses pembayaran. Setelah admin melakukan proses pembayaran uang yang diterima akan dikurangi total sehingga akan didapatkan uang kembalian yang akan ditampilkan pada halmaan di atas.

#### Validasi Pesanan

Setelah pengguna melakukan pesanan, admin grup harus melakukan validasi tergadap pesanan tersebut. Sehingga status pesanannya menjadi validasi dan progressnya menjadi on progress.

|  |
| --- |
|  |

Gambar 2. 45 Halaman List Validasi Pesanan

Gambar 2.45 merupakan tampilan dari halaman validasi pesanan dari sisi admin grup. Setelah user menambah pesanan maka admin dapat melaukan validasi terhadap pesanan tersebut. Admin group dapat mengklik tombol validate utnuk menampilkan fitur inline edit untuk melakukan validasi seperti pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 46 Inline Edit Validasi Pesanan

Gambar 2.46 merupakan tampilan dari fitur inline edit dari halaman validasi pesanan. Admin grup dapat melakukan perubahan pada field valid dengan mengubah valuenya menjadi valid atau tetap membiarkannya not valid. Setelah divalidasi maka progress dari setiap detail akan diubah menjadi on progress dan akan muncul pada halaman list update progress pesanan seperti pada gambar berikut.

#### Update Progress Pesanan

Setelah pesanan tervalidasi, admin grup dapat melakukan update progress pesanan dengan mengubah progress yang awalnya on progress menjadi ready. Hali ini berarti pesanan sudah siap unutk disajikan ke pelanggan.

|  |
| --- |
|  |

Gambar 2. 47 Halaman Update Progress Pesanan

Gambar 2.47 merupakan tampilan dari halaman list update progress laporan. Pada halaman ini muncul setiap detail dari pesanan yang saudah divalidasi sebelumnya. Untuk melakukan update progress admin grup dapat mengklik tombol update progress dan akan muncul fitur inline edit seperti gambar berikut.

|  |
| --- |
|  |

Gambar 2. 48 Inline Edit Halaman Update Progress Pesanan

Gambar 2.48 merupakan tampilan dari fitur inline edit pada halaman update progress pesanan. Field progress dapat diubah menjadi ready oleh admin grup. Setelah semua detail pesanan sudah ready admin group dapat melanjutkan ke proses pembayaran seperti pada gambar berikut.

### Panduan Pengoperasian User

User biasa pada sistem ini dapat melakukan berbagai hal diantaranya mendaftarkan akun baru, bergabung dengan grup, membuat personal channel, membuat personal agenda, membuat pesanan tanpa checkin, melakukan checkin, membuat pesanan setelah checkin, memberikan ulasan pesanan, dan memberikan ulasan checkin.

#### Mandaftarkan Akun Baru

Penguna dapat melakukan pendaftaran terlebih dahulu sebelum dapat menggunkan sistem ini. Tampilan dan alur ini terdiri dari proses registrasi dan login. Berikut merupakan tampilan dari halaman registrasi.

|  |
| --- |
|  |

Gambar 2. 49 Halaman Registrasi

Gambar 2.49 merupakan tampilan dari registrasi. Terdapat lima inputan yang harus dimasukan oleh user pada login yaitu Email, Password, Nama, Negara dan Provinsi untuk dapat melakukan login. User yang telah memasukkan kelima inputan tersebut dapat melakukan login dengan mengklik tombol “Daftar” dan selanjutnya akan dialihkan ke halaman pendaftaran berhasil dengan arahan untuk memeriksa email untuk menverifikasi akun.

|  |
| --- |
|  |

Gambar 2. 50 Pemberitahuan untuk Cek Email

Gambar 2.50 merupakan tampilan dari halaman pendaftran berhasil dan arahan untuk memeriksa email unutk mendapatkan email aktifasi atau validasi akun personal. Setelah tampil halaman tersebut pengguna dapat membuka email dan mencari pesan seperti yang ditampilkan pada gambar dibawah ini.

|  |
| --- |
|  |

Gambar 2. 51 Email yang Dikiriman Oleh Sistem

Gambar 2.51 merupakan tampilan dari email yang dikirimkan sistem ke alamat email pengguna. Pada email tersebut terdapat informasi akunpersonal pengguna dan link untuk melakukan validasi akun. Setelah masuk ke link tersebut pengguna akan diarahkan ke sistem dengan tampilan pendaftran berhasil dengan tombol login di bawahnya.

|  |
| --- |
|  |

Gambar 2. 52 Halaman Login

Gambar 2.52 merupakan tampilan dari login. Terdapat dua inputan yang harus dimasukan oleh user pada login yaitu Email dan Password user untuk dapat melakukan login. User yang telah memasukkan Email dan Password dapat melakukan login dengan mengklik tombol “Masuk”. User yang login akan langsung diarahkan ke halaman utama dari sistem dengan role pengguna sebagai user biasa.

|  |
| --- |
|  |

Gambar 2. 53 Menu Utama Sisi User

Gambar 2.53 merupakan tampilan dari halmaan utama dari sistem dengan role pengguna sebagai user biasa. Pengguna dengan role user dapat melakukan beberapa hal diantaranya membuat dan review pesanan, melakukan checkin, bergabung dengan grup, membuat grup, dan membuat agenda.

#### Bergabung dengan Grup

Sebelum user dapat melakukan checkin, user aharus bergabung dengan salah satu grup terlebuh dahulu. Untuk bergabung dengan salah satu grup user dapat menuju ke menu keanggotaan grup.

|  |
| --- |
|  |

Gambar 2. 54 Halaman Awal Keanggotaan Grup

Gambar 2.54 merupakan tampilan dari halaman list keanggotaan grup. Pada halaman ini aka mucul semua grup yang diikuti oleh user baik yang sudah valid maupun belum valid. User dapat mengklik tombol Tambah Baru untuk join ke salah satu grup yang ada di sistem.

|  |
| --- |
|  |

Gambar 2. 55 Halman Tambah Keanggotaan Grup Baru

Gambar 2.55 merupakan tampilan dari form tambah baru dari keanggotaan grup. Terdapat satu inputan yang harus dipilih oleh user pada halaman ini yaitu memilih nama grup yang ingin dimasuki. Setalah memilih nama grup user dapat mengklik tombol Simpan untuk masuk ke grup

|  |
| --- |
|  |

Gambar 2. 56 Halaman Setelah Join Grup Public

Gambar 2.56 merupakan tampilan dari halaman list keanggotaan grup. Pada gambar diatas terdapat satu buah data yang merupakan data yang dimasukkan oleh user sebelumnya. Status validasi dari data tersebut saudah valid karena grup yang dimasukki memiliki tipe public, maka user tidak perlu memasukkan token.

|  |
| --- |
|  |

Gambar 2. 57 Halaman Memasukkan Token Grup

Gambar 2.57 merupakan tampilan dari halman menambahkan token dari grup dengan tipe token oleh user. Pada gambar diatas dilakukan iput pada filed token grup dengan value yang sama seperti token yang dimasukkan oleh admin grup.

|  |
| --- |
|  |

Gambar 2. 58 Halaman Memasukkan Token Pribadi

Gambar 2.58 merupakan tampilan dari halman menambahkan group indentifier dan token dari grup dengan tipe lookup oleh user. Pada gambar diatas dilakukan input pada filed pengenal group dan token pribadi dengan value yang sama seperti yang dimasukkan oleh admin grup sebelumnya.

#### Membuat Personal Channel

Personal channel adalah channel yang dimiliki oleh user. Maksud dari channel adalah alamat email atau id telegram sesusai dengan pilihan jenis channel pilihan user. Berikut merupakan cara mendaftarkan channel yang dimiliki.

|  |
| --- |
|  |

Gambar 2. 59 Daftar Channel

Gambar 2.59 merupakan tampilan list channel yang sudah didaftarkan. Terlihat terdapat list channel dari nama user dan jenis channelnya. Channel ini berguna untuk menerima pesan dari agenda personal maupun agenda grup. Untuk membuat channel baru maka klik tombol add new di pojok kiri atas halaman web.

|  |
| --- |
|  |

Gambar 2. 60 Tampilan Menambah Channel

Gambar 2.60 merupakan tampilan awal pembuatan personal channel. Terlihat terdapat inputan channel name yang merupkan jenis channel yang dimiliki user dan ingin didaftarkan, terdapat beberapa pilihan seperti telegram, email, whatsapp, dll. Kemudian terdapat inputan chat id yang merupakan alamat email jika memilih email, atau id telegram, atau nomer telepon whatsapp. Setelah mengklik tombol save, maka pendaftaran channel berhasil dan terdapat pada list channel.

#### Membuat Personal Agenda

Agenda personal adalah agenda yang dibuat oleh personal dari aplikasi reminder. Agenda ini dibuat sebagai reminder untuk kepentingan pribadi.

|  |
| --- |
|  |

Gambar 2. 61 Tampilan Daftar Agenda Personal

Gambar 2.61 merupakan tampilan awal dari agenda personal. Terlihat terdapat daftar agenda yang sudah dibuat oleh suatu personal. Terdapat nama personal, loop type, tanggal kirim, konten pesan, lampiran, jumlah repeat, dan repeat type. Untuk membuat personal agenda dapat meng-click tombol add new pada bagian atas kiri halaman web.

|  |
| --- |
|  |

Gambar 2. 62 Halaman Pembuatan Personal Agenda Tanpa Memiliki Channel

Gambar 2.62 merupakan tampilan saat ingin membuat agenda personal namun belum memiliki channel. Saat seorang personal ingin membuat agenda, harus dipastikan sudah mendaftarkan channelnya terlebih dahulu. Karena jika belum maka tidak dapat membuat agenda personal. Channel yang dimaksud adalah alamat email atau id telegram.

|  |
| --- |
|  |

Gambar 2. 63 Tampilan Awal Pembuatan Agenda Personal

Gambar 2.63 merupakan tampilan awal pembuatan agenda personal. Terdapat 3 inputan awal yakni message content untuk menuliskan pesan yang ingin disampaikan, kemudian ada inputan attachment yakni untuk memberikan lapiran dalam bentuk file, kemudian terdapat inputan loop type untuk menentukan perulangan dari agenda tersebut. Dalam loop type terdapat pilihan real time untuk agenda pada hari itu atau ondate dengan tambahan inputan tanggal yang akan dijelaskan selanjutnya.

|  |
| --- |
|  |

Gambar 2. 64 Contoh Pengisian Pembuatan Agenda Personal Real Time

Gambar 2.64 merupakan contoh saat mengisi personal agenda. Terlihat message content diisi dengan pesan “peringatan”, kemudian ditambahkan attachment berupa lampiran file excel, serta pemilihan loop type real time yang mana saat di clik nya tombol save maka akan langsung membuat agenda pada hari tersebut. Selanjutkan akan dijelaskan jika memilih loop type on date.

|  |
| --- |
|  |

Gambar 2. 65 Contoh Pengisian Pembuatan Agenda Personal Ondate

Gambar 2.65 merupakan tampilan saat user memilih loop type on date. Terlihat saat memilih on date maka akan tampil inputan tambahan berupa loop value yang merupakan tanggal pengiriman agenda, kemudia terdapat juga repeat yakni jumlah pengulangan pengiriman agenda yang diinginkan, jika diisi 0 maka tidak melakuka perulangan dan ketika di click tombol save maka akan membuat agenda pada tanggal berdasarkan loop value tersebut. Namun jika diisi dengan angka lebih dari 0 maka akan terdapat inputan tambahan yang akan dijelaskan selanjutnya.

|  |
| --- |
|  |

Gambar 2. 66 Contoh Pengisian Pembuatan Agenda Personal Repeat Type Bydate

Gambar 2.66 merupakan tampilan saat mengisi nilai repeat dengan angka lebih dari 0. Terlihat terdapat tambahan repeat type yang memiliki beberapa pilihan yakni daily untuk melakukan perulang ditambah 1 hari dari tanggal loop value sebanyak nilai inputan repeat. Kemudian terdapat pilihan weekly untuk melakukan perulang ditambah 1 minggu dari tanggal loop value sebanyak nilai inputan repeat. Kemudian terdapat pilihan monthly untuk melakukan perulang ditambah 1 bulan dari tanggal loop value sebanyak nilai inputan repeat. Kemudian terdapat pilihan yearly untuk melakukan perulang ditambah 1 tahun dari tanggal loop value sebanyak nilai inputan repeat. Yang mana jika memilih pilihan repeat type dari salah satu pilihan tersebut, saat user meng-click tombol save maka akan dibuat agenda perulangan sebanyak inputan yang telah diisi tersebut. Terakhir terdapat pilihan bydate yang akan dijelaskan selanjutnya.

|  |
| --- |
|  |

Gambar 2. 67 Inputan Tanggal Dari Repeat Type Bydate Personal Agenda

Gambar 2.67 merupakan tampilan saat user memilih repeate type bydate pada pembuatan personal agenda. Terlihat terdapat nama dari personal agenda yakni peringatan yang telah dibuat sebelumnya, kemudian terdapat inputan dodate yang berupa tanggal yang ingin dilakukan perulangan agenda. Jumlah inputan dodate tergantung dari nilai inputan repeat sebelumnya. Saat user meng-click save maka akan dibuat personal agenda dengan perulangan sebanyak nilai repeat, serta tanggal sesuai dengan dodate.

|  |
| --- |
|  |

Gambar 2. 68 Daftar Personal Agenda Berhasil Dibuat

Gambar 2.68 merupakan tampilan daftar dari personal agenda yang sudah dibuat. Terlihat terdapat tabel dengan isi dari personal agenda serta repeate bydate dengan tanggal yang diisi pada inputan dodate.

#### Membuat Pesanan Tanpa Checkin

User dapat melakukan pesanan tanpa harus melakukan checkin terleih dahulu. Hal ini dapat terjadi karena terdapat produk yang memiliki tipe order public dan member. Berikut merupakan tampilan dari halaman list Pesanan Anggota Grup dari sisi user.

|  |
| --- |
|  |

Gambar 2. 69 Halaman List Pesanan Anggota Grup

Gambar 2.69 merupakan tampilan dari halaman list pesanan anggota grup. Pada halaman ini akan tampil semua pesanan yang ada di dilakukan oleh user. User juga dapat menambahkan pesanan baru dengan mengklik tombol Tambah Baru. Tampilan dari halaman tambah pesanan baru dapat dilihat pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 70 Halaman Tambah Pesanan Anggota Grup Public

Gambar 2.70 merupakan tampilan dari tambah pesanan baru dari grup yang memiliki tipe order public sehingga user tidak perlu join grup untuk dapat melakukan pesanan. Terdapat dua inputan yang harus dimasukan oleh user pada memilih grup dan memilih mata uang.

|  |
| --- |
|  |

Gambar 2. 71 Popup Tambah Detail Pesanan Grup Public

Gambar 2.71 merupakan tampilan dari tambah detail pesanan baru dari grup yang memiliki tipe order public sehingga user tidak perlu join grup untuk dapat melakukan pesanan. Terdapat dua inputan yang harus dimasukan oleh user pada memilih produk dan memasukkan jumlah.

|  |
| --- |
|  |

Gambar 2. 72 Halaman Tambah Pesanan Anggota Grup

Gambar 2.72 merupakan tampilan dari tambah pesanan baru dari grup yang sudah dimasukki oleh user. Terdapat dua inputan yang harus dimasukan oleh user pada memilih grup dan memilih mata uang. Selanjutnya user dapat memasukkan detail pesanan pada pesanan tersebut.

|  |
| --- |
|  |

Gambar 2. 73 Popup Tambah Detail Pesanan Grup Tipe Member

Gambar 2.73 merupakan tampilan dari tambah detail pesanan baru dari grup yang sudah dimasukki oleh user. Terdapat dua inputan yang harus dimasukan oleh user pada memilih produk dan memasukkan jumlah. Pada filed Nama produk terlihat hanya muncul 2 produk dari 3 produk yang dimasukkan oleh admin sebelumnya. Hal ini dikarenakan 1 produk bertipe order member checkin sehingga user haru checkin sebelum dapat melakukan pemesanan produk tersebut.

#### Melakukan Checkin

User dapat melakukan checkin untuk dapat melakukan pesanan terhadap produk dengan tipe member checkin. Untuk melakukan checkin user dapat menuju ke menu member checkin.

|  |
| --- |
|  |

Gambar 2. 74 Halaman Tambah Checkin Anggota

Gambar 2.74 merupakan tampilan dari halaman tambah checkin baru dari sisi user. Terdapat emapt kolom ynag harus diisi oleh user yaitu memilih grup, id room, tanggal checkin dan tanggal checkout. Setalah melakukan checkin admin harus memvalidasi terlebih dahulu sebelum user dapat melakukan pesanan produ dengan tipe order checkin.

#### Membuat Pesanan Setelah Checkin

Setelah user melakukan checkin, user dapat melakukan pemesanan terhadap produk dengan tipe order member checkin. Berikut merupakan tampilan dari memesan produk dengan tipe member checkin.

|  |
| --- |
|  |

Gambar 2. 75 Popup Tambah Detail Pesanan Grup Tipe Member Checkin

Gambar 2.75 merupakan tampilan dari tambah detail pesanan baru setelah user melakukan checkin. Pada field nama produk sudah muncul semua produk yang telah dimasukkan oleh admin grup sebelumnya baik yang bertipe order public, member, maupun member checkin.

|  |
| --- |
|  |

Gambar 2. 76 Halaman Pesanan Anggota Grup Setelah Memesan

Gambar 2.76 merupakan tampilan dari halaman pesanan anggota grup dan detail pesanan anggota grup yang merupakan detail dari salah satu data pesanan anggota grup. Pada gambar di atas ditampilkan data pesanan dan detail dari pesanan yang sudah ditambahkan sebelumnya.

#### Memberikan Ulasan Pesanan

Setelah melakukan pemesana dan sudah selesai diproses oleh admin grup, pengguna dapat memberikan ulasan terhadap pesanan yang telah terselesaikan. Tampilan dari menu ulasan pesanan dapat dilihat pada gamabr berikut.

|  |
| --- |
|  |

Gambar 2. 77 Halaman List Ulasan Pesanan

Gambar 2.77 merupakan tampilan dari halaman list ulasan pesanan. Pada halaman ini akan muncul semua data pesanan yang sudah selesai diproses dan belum direview. Untuk dapat menambahkan review user dapat klik pada tombol Add Review maka akan mucul halaman tambah review seperti pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 78 Halaman Tambah Ulasan Pesanan

Gambar 2.78 merupakan tampilan dari halaman menambahkan ulasan dan penilaian pada pesanan yang telah dipilih sebelumnya. Pada halaman ini user dapat memasukkan ulasan dan penilainnya terhadap pesanan yang telah terselesaikan tersebut.

#### Memberikan Ulasan Checkin

Setelah melakukan checkin dan sudah divalidasi oleh admin grup, pengguna dapat memberikan ulasan terhadap checkin yang telah terselesaikan. Tampilan dari menu ulasan chcekin dapat dilihat pada gamabr berikut.

|  |
| --- |
|  |

Gambar 2. 79 Halaman List Ulasan Checkin

Gambar 2.79 merupakan tampilan dari halaman list ulasan checkin. Pada halaman ini akan muncul semua data checkin yang sudah selesai diproses dan belum direview. Untuk dapat menambahkan review user dapat klik pada tombol Add Review maka akan mucul halaman tambah review seperti pada gambar berikut.

|  |
| --- |
|  |

Gambar 2. 80 Halaman Tambah Ulasan Checkin

Gambar 2.80 merupakan tampilan dari halaman menambahkan ulasan dan penilaian pada checkin yang telah dipilih sebelumnya. Pada halaman ini user dapat memasukkan ulasan dan penilainnya terhadap checkin yang telah terselesaikan tersebut.