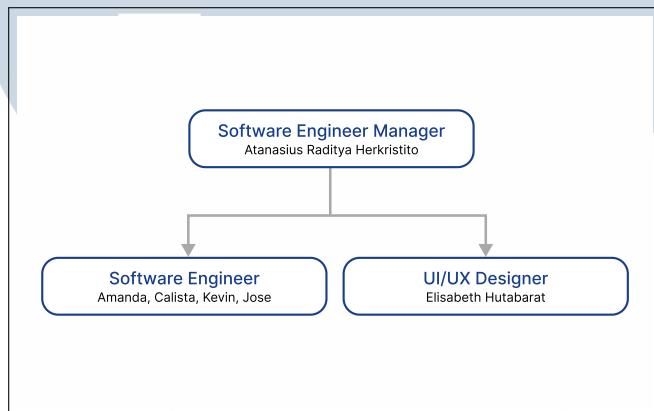


BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama pelaksanaan kegiatan magang di PT Visi Karya Nusantara, posisi yang ditempati adalah Software Engineer di bawah pengawasan dan bimbingan Atanasius Raditya Herkristito. Tugas utama selama magang adalah mengembangkan *Human Resource Information System (HRIS)*, yang mencakup perancangan dan implementasi antarmuka untuk meningkatkan efisiensi sistem dalam pengelolaan data kepegawaian.



Gambar 3.1. Struktur organisasi Pengembang

Dalam pengembangan HRIS, tim terdiri dari satu *Software Engineer Manager*, satu *UI/UX Designer*, dan empat *Software Engineer*. Selama proses pengembangan ditempatkan pada sisi *frontend* yang berkoordinasi secara langsung dengan *Software Engineer* pada sisi *backend*, *UI/UX Designer*, serta *Software Engineer Manager*, terutama pada setiap pelaksanaan *sprint*. Komunikasi dan koordinasi tim dilakukan melalui aplikasi Discord sebagai media utama. Tinjauan progres dan *code review* dilaksanakan setiap minggu, tepatnya pada hari Kamis. Selain itu, dilakukan sesi evaluasi bersama seluruh tim untuk memberikan *feedback* terhadap perkembangan proyek guna meningkatkan kualitas pengembangan pada tahap berikutnya.

3.2 Tugas yang Dilakukan

Selama Pelaksanaan magang di PT Visi Karya Nusantara, tugas utama yang dikerjakan adalah mengembangkan *Human Resource Information System* (HRIS), yang direncanakan menjadi produk resmi dan digunakan oleh perusahaan. Adapun tugas yang dilakukan meliputi:

1. Membangun *frontend* pada modul *contract* dan *payroll system*
2. Mengembangkan sistem absensi yang mencakup fitur *clock in*, *clock out*, dan *lunch* pada modul *attendance*.
3. Mengembangkan modul *Event* yang mencakup proses pembuatan kegiatan serta pengumuman (announce) secara langsung pada *dashboard* pengguna.
4. Melakukan manual *testing* terhadap HRIS.

3.3 Uraian Pelaksanaan Magang

Tabel 3.1 di atas merupakan uraian pelaksanaan magang yang dilaksanakan di PT Visi Karya Nusantara. Selama 18 minggu, terlibat dalam pengembangan *frontend* dimulai dari proses *onboarding* dan mempelajari sistem HR. Tahap awal mencakup pembuatan modul *Contract*, melakukan integrasi API, serta melakukan manual *testing* untuk memastikan fungsionalitas berjalan dengan baik. Selanjutnya, dilakukan pengembangan modul *Payroll System* yang kemudian diintegrasikan dengan modul *Contract*, dilanjutkan dengan proses *manual testing* untuk memastikan kedua modul bekerja secara selaras.

Pada tahap berikutnya, dikembangkan sistem *Attendance* yang terhubung dengan modul *Contract*, termasuk penambahan fitur *Lunch Attendance*. Setelah itu, dilakukan pengembangan modul *Event* beserta fitur-fiturnya, serta integrasi API dan proses *manual testing*. Tahap akhir meliputi pengecekan keseluruhan halaman dan fitur, melakukan *bug fixing*, dan menjalankan proses *deployment* ke *production* hingga sistem siap digunakan sebagai produk akhir.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	<i>Onboarding</i> dan mempelajari <i>flow</i> sistem HRIS.
2	Melalukan <i>research</i> terkait integrasi modul <i>Contract</i> pada HRIS
3	Mengerjakan UI modul <i>Contract</i> .
4	Melakukan integrasi API modul <i>Contract</i> .
5	Melakukan manual <i>testing</i> pada modul <i>Contract</i> .
6	Mengerjakan UI modul <i>Payroll System</i> .
7	Melakukan integrasi API modul <i>Payroll System</i> dan pada <i>Contract Input</i> .
8	Melakukan manual <i>testing</i> pada modul <i>Payroll System</i> .
9	Melakukan perbaikan bug (<i>bug fixing</i>) dan proses <i>deployment</i> .
10	Melakukan <i>research</i> terkait proses <i>attendance</i> pada HRIS
11	Melakukan integrasi API fitur <i>Attendance</i>
12	Mengerjakan UI fitur <i>Lunch Attendance</i>
13	Melakukan integrasi <i>Attendance</i> dengan <i>Contract Schedule</i>
14	Melakukan manual <i>testing</i> pada fitur <i>Attendance</i>
15	Melakukan pengembangan UI modul <i>Event</i>
16	Melakukan integrasi API modul <i>Event</i>
17	Melakukan manual <i>testing</i> pada modul <i>Event</i>
18	Melakukan perbaikan bug (<i>bug fixing</i>) dan proses <i>deployment</i>

3.4 Perancangan

Sub bab ini menjelaskan proses perancangan aplikasi *Human Resource Information System* (HRIS), yang mencakup *User Requirement*, *Sitemap*, dan *Flowchart*.

3.4.1 *User Requirement*

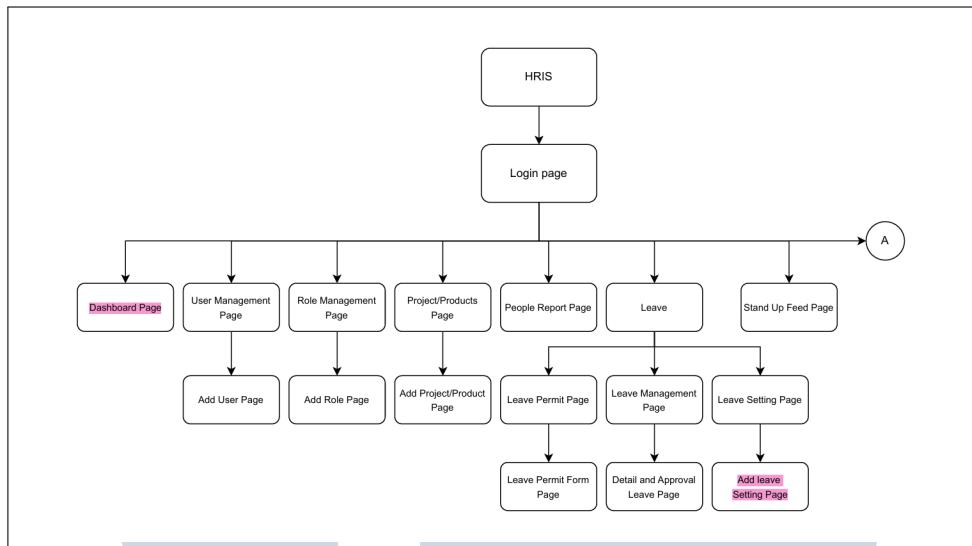
Sebelum memulai pengembangan, *Software Engineer Manager* memberikan arahan terkait tugas yang akan dilakukan dalam pengembangan *Human Resource Information System* (HRIS). Aspek yang perlu dikembangkan di antaranya adalah sebagai berikut:

1. Mengembangkan halaman *Contract* untuk pengelolaan kontrak karyawan, meliputi pembuatan, pengeditan, dan penghapusan kontrak. Selain itu, tersedia fitur *approval*, *rejection*, *cancel*, dan *termination* sesuai dengan status kontrak. Karyawan juga dapat melihat detail kontrak mereka melalui halaman *Contract User*.
2. Mengembangkan halaman *Payroll Setting* untuk pengaturan item gaji dan tunjangan yang digunakan pada *Contract Input*. Halaman ini terdiri dari *Payroll Setting List* yang menampilkan seluruh item payroll yang tersedia, serta *Payroll Setting Input* untuk menambahkan item baru.
3. Mengembangkan fitur *Attendance System* pada halaman *Dashboard* untuk mengatur proses absensi karyawan. Pembaruan dilakukan dengan menambahkan fitur *clock in* bagi karyawan dengan sistem kerja *shift* maupun *fixed hour*, yang disesuaikan berdasarkan jadwal dan jam kerja pada kontrak masing-masing.
4. Mengembangkan modul *Event* yang mencakup pembuatan kegiatan dan pengumuman (*announce*) secara langsung pada *dashboard* pengguna. Fitur ini memungkinkan penambahan deskripsi kegiatan, dokumen pendukung, serta pembaruan data secara *real-time* agar pengguna dapat dengan mudah memantau dan mengelola jadwal kegiatan yang akan datang.

3.4.2 *Sitemap*

Perancangan struktur navigasi melalui *sitemap* sangat penting dalam pengembangan *Human Resource Information System* (HRIS) untuk memastikan alur kerja pengguna tetap terorganisir dan efisien. *Sitemap* ini berfungsi sebagai peta visual yang menggambarkan hierarki informasi serta hubungan antarhalaman.



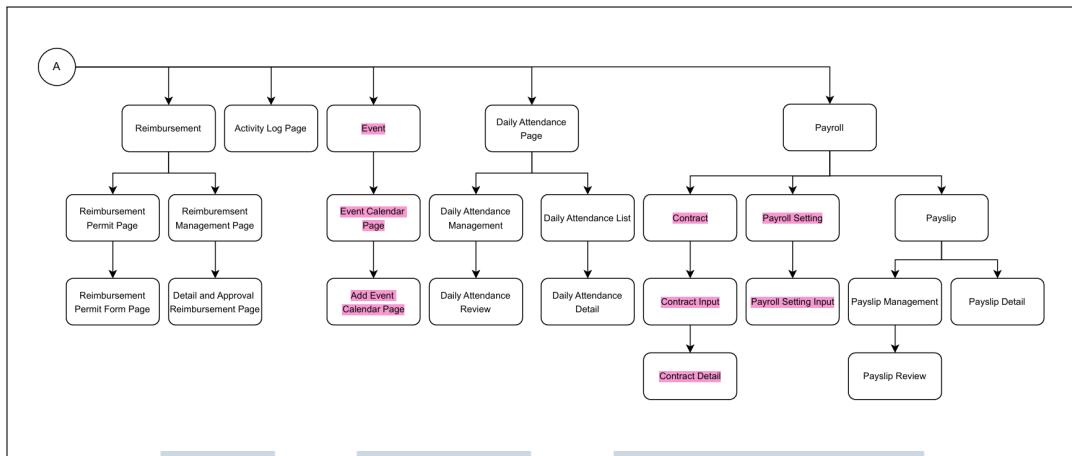


Gambar 3.2. Sitemap A Human Resource Information System

Gambar 3.2 menunjukkan *sitemap* bagian pertama yang mencakup alur autentikasi serta beberapa manajemen data inti dalam sistem HRIS. Proses operasional dimulai dari entitas HRIS menuju halaman *login*, di mana setiap pengguna diwajibkan melakukan proses autentikasi identitas untuk keamanan akses data perusahaan. Setelah proses *login* berhasil, sistem akan mengarahkan pengguna ke halaman *dashboard* yang merupakan halaman utama untuk mengakses berbagai modul administratif seperti *User Management*, *Role Management*, hingga *People Report Page*.

Sebagai kelanjutan dari struktur sebelumnya, bagian kedua dari *sitemap* ini berfokus pada modul-modul operasional yang lebih spesifik dan kompleks dalam mendukung kegiatan harian perusahaan. Bagian ini mencakup fungsionalitas pengelolaan keuangan, absensi, hingga pengaturan agenda internal tim. Diagram sitemap ini memvisualisasi hubungan antarhalaman fitur-fitur baru yang dikembangkan selama masa magang.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3. Sitemap B Human Resource Information System

Gambar 3.3 memvisualisasikan kelanjutan navigasi sistem yang mencakup modul-modul utama seperti *Reimbursement*, *Event*, *Daily Attendance*, dan *Payroll*. Fokus utama pengembangan dalam laporan magang ini meliputi modul yang diberikan tanda *highlight*, yaitu modul *Dashboard*, *Leave Setting Input*, *Contract*, *Event*, serta *Payroll Setting*. Struktur navigasi ini dirancang untuk memudahkan pengguna dalam mengakses fitur yang dibutuhkan secara terorganisir dan efisien.

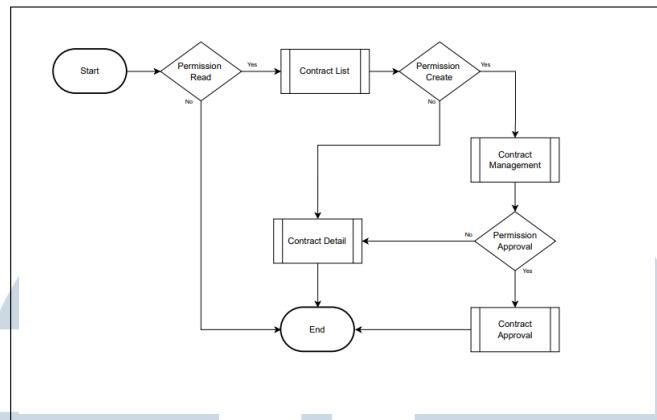
3.4.3 Flowchart

Pada bagian ini ditampilkan *flowchart* yang menggambarkan alur kerja setiap modul dalam sistem secara menyeluruh. Diagram ini berfungsi untuk memberikan pemahaman yang lebih jelas mengenai tahapan proses, keputusan, serta interaksi antar komponen yang terjadi selama sistem berjalan. Dengan adanya visualisasi ini, pembaca dapat melihat bagaimana data diproses, bagaimana logika sistem bekerja, serta bagaimana setiap fitur saling terhubung satu sama lain. Alur lengkap dari proses yang berlangsung pada modul tersebut dapat dilihat sebagai berikut:

U
N
I
V
E
R
S
T

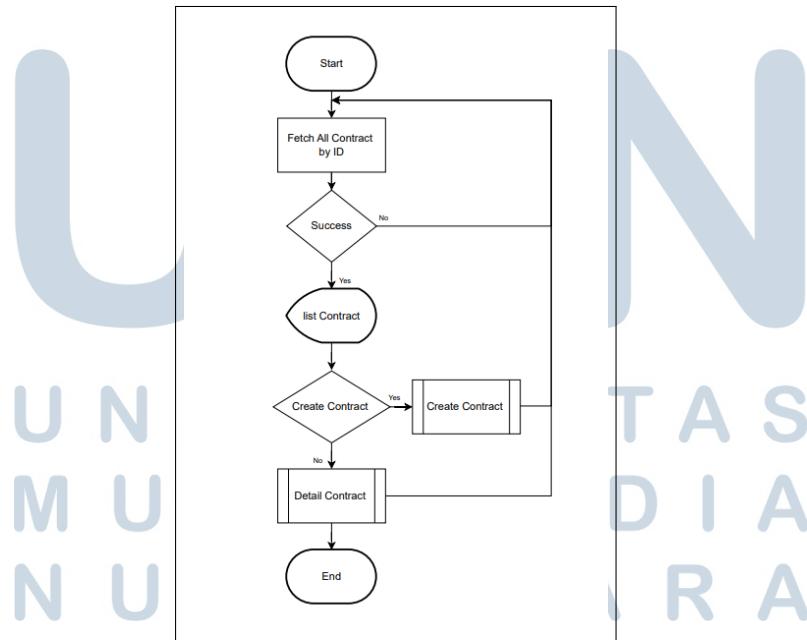
M
U
L
T
I
M
E
D

N
U
S
A
N
T
R
A



Gambar 3.4. *Flowchart Contract Permission Page*

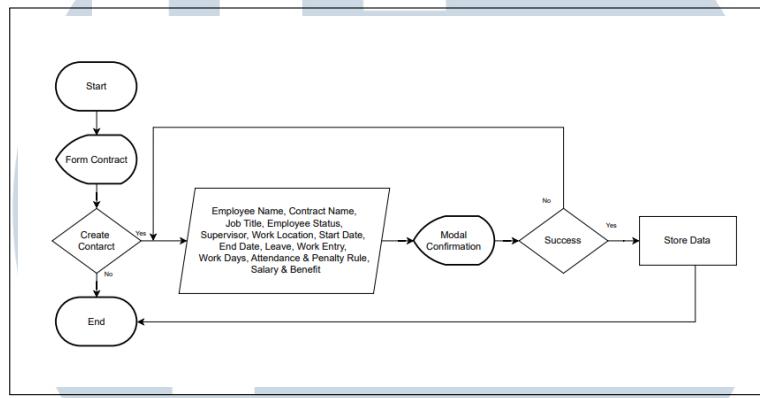
Gambar 3.4 menunjukkan *flowchart* pembagian halaman berdasarkan *permission role* yang diberikan pada halaman *Contract*. Pada proses ini, sistem terlebih dahulu melakukan pengecekan terhadap akun pengguna yang sedang melakukan *login* untuk mengidentifikasi *role* yang dimiliki. Berdasarkan *role* tersebut, sistem kemudian menentukan izin (*permission*) yang sesuai, sehingga hanya fitur atau halaman tertentu yang dapat diakses oleh pengguna. Mekanisme ini diterapkan untuk menjaga keamanan data kontrak serta memastikan setiap pengguna hanya dapat mengakses fungsi sesuai dengan tanggung jawabnya.



Gambar 3.5. *Flowchart Contract List*

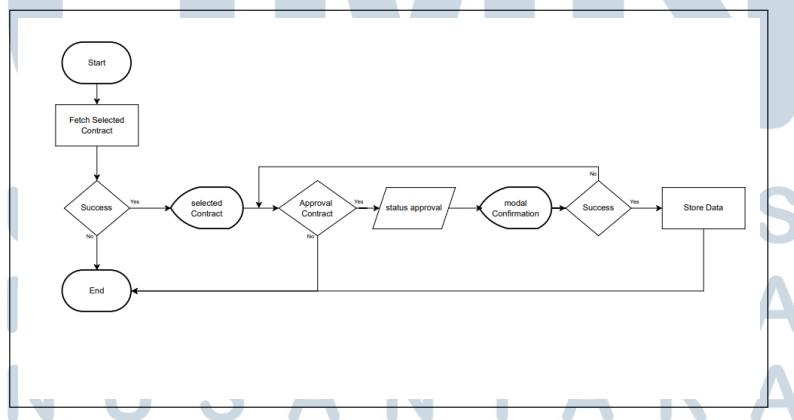
Gambar 3.5 memperlihatkan *flowchart* pada halaman *Contract List*. Proses

dimulai dengan pemanggilan *GET API* untuk menampilkan seluruh data *contract* yang telah tersimpan di dalam sistem. Selanjutnya, pengguna yang memiliki *permission create* dapat melakukan pembuatan *contract* baru, sedangkan pengguna dengan *permission approval* dapat melakukan proses persetujuan (*approval*) terhadap *contract* sesuai dengan peran yang dimiliki.



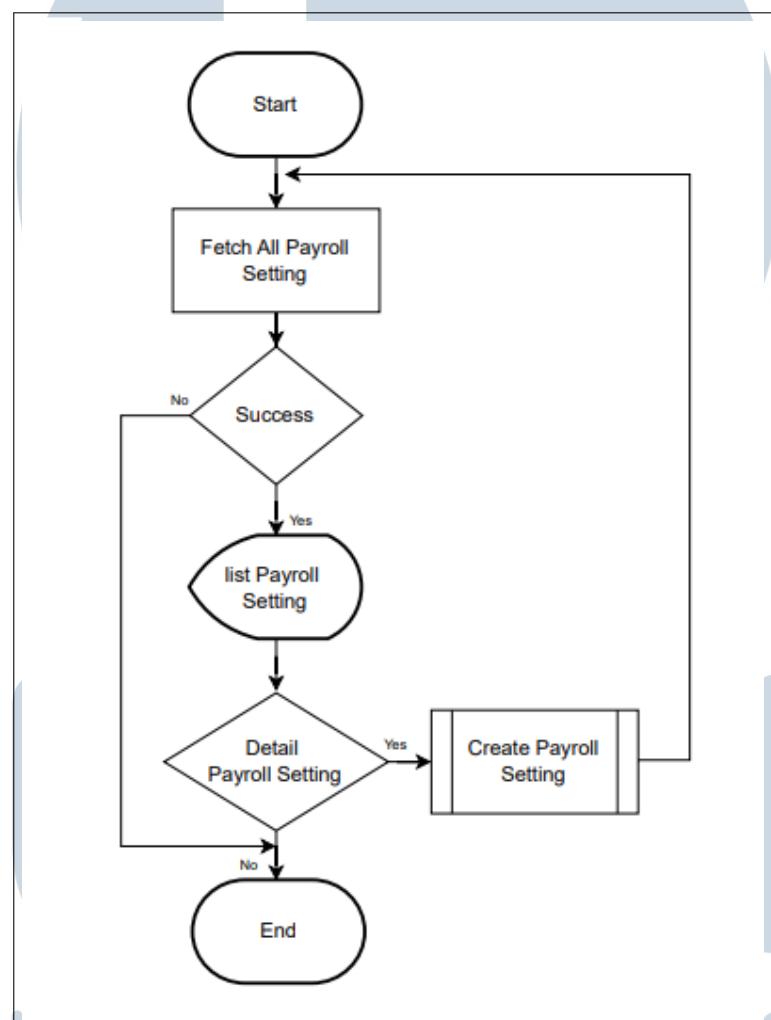
Gambar 3.6. *Flowchart Contract Create*

Gambar 3.6 merupakan *flowchart* pada halaman *Contract Create*. Proses dimulai dengan menampilkan *form field* yang digunakan untuk mengisi data perjanjian karyawan. Pada halaman ini, pengguna dapat melakukan kustomisasi terhadap data *penalty & rule*, jadwal kerja karyawan, serta komponen gaji yang disesuaikan dengan kebutuhan. Setelah seluruh data diisi, sistem akan menampilkan *modal konfirmasi*, dan setelah pengguna memberikan persetujuan, data kontrak akan disimpan ke dalam *database*.



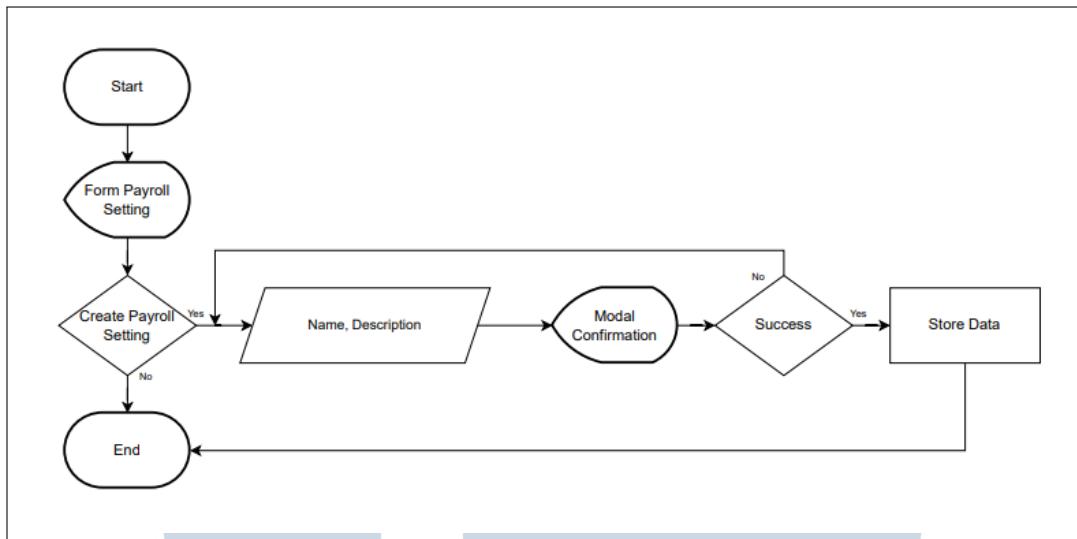
Gambar 3.7. *Flowchart Contract Detail Approval*

Gambar 3.7 menunjukkan *flowchart* pada halaman *Contract Detail Approval*. Proses dimulai dengan pemanggilan GET API untuk mengambil data kontrak yang dipilih. Jika pengguna memiliki *permission approval*, maka dapat melakukan *accept* atau *reject* terhadap kontrak. Setelah itu, ditampilkan modal konfirmasi, dan kontrak disimpan ke *database* serta dikirim kepada pengguna untuk mengaktifkan masa kontrak.



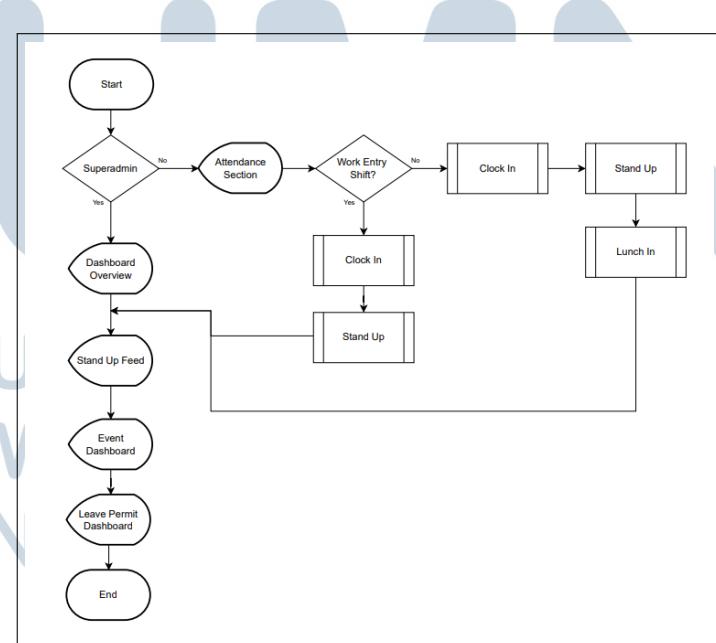
Gambar 3.8. *Flowchart Payroll List*

Gambar 3.8 adalah *flowchart* pada halaman *Payroll List*. Proses dimulai dengan pemanggilan GET API untuk menampilkan seluruh item *Payroll Setting*. Pengguna dapat melihat seluruh item *Payroll Setting* apabila mendapatkan *permission read*. Pengguna dapat membuat item *payroll* baru apabila mendapatkan *permission create* berdasarkan role yang dimiliki.



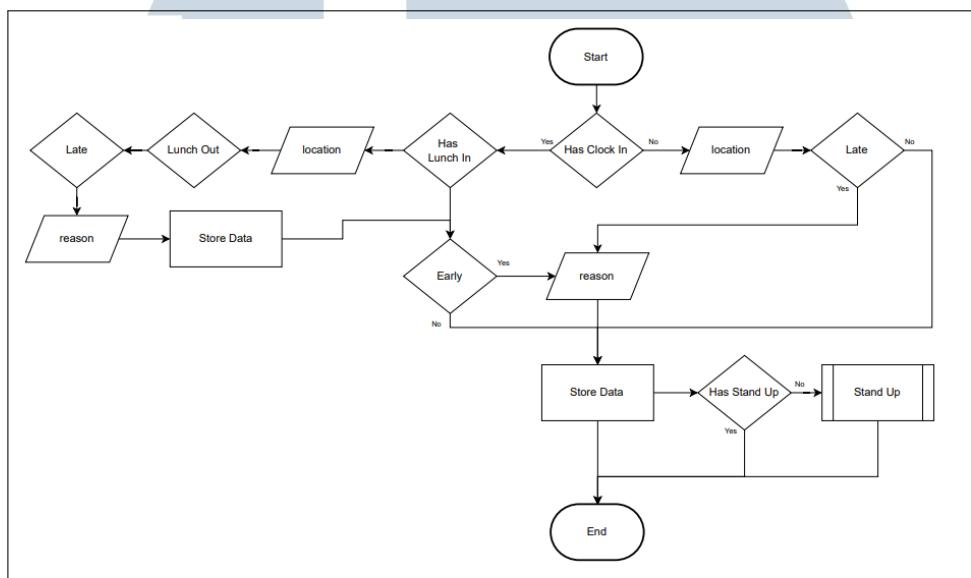
Gambar 3.9. *Flowchart Payroll Create*

Gambar 3.9 merupakan *flowchart* pada halaman *Payroll Create*. Proses dimulai dengan menampilkan *form field* untuk pengisian *item payroll* oleh *admin*, di mana data yang diperlukan mencakup *item payroll* dan *description*. Setelah *admin* melengkapi data, sistem akan memunculkan sebuah modal konfirmasi pengajuan untuk melakukan validasi. Jika konfirmasi disetujui, maka data akan disimpan ke dalam *database* agar dapat digunakan kembali secara fleksibel dalam modul *payslip* dan *contract* karyawan.



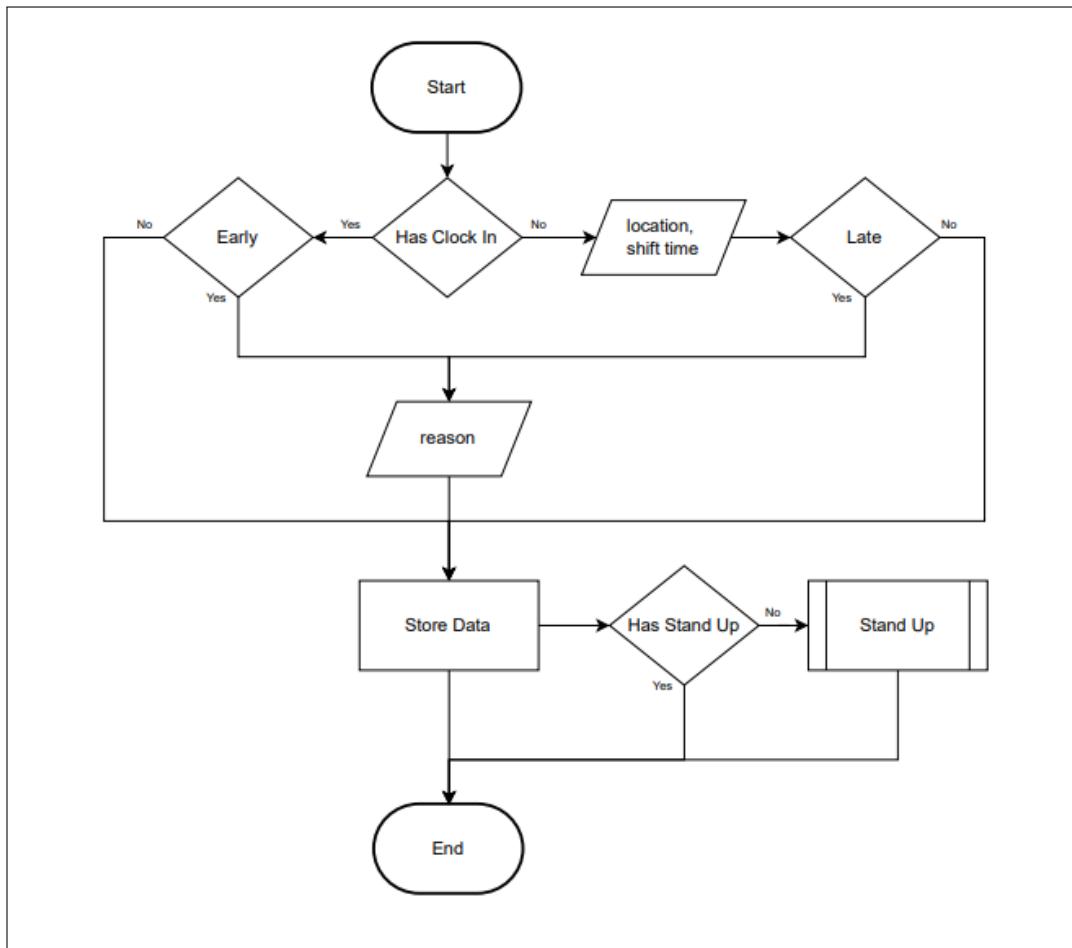
Gambar 3.10. *Flowchart Attendance System*

Gambar 3.10 memperlihatkan *flowchart* pada halaman *Dashboard Permission*. Apabila pengguna merupakan *superadmin*, maka sistem akan menampilkan *dashboard overview* yang berisi data-data internal. Jika bukan, sistem akan memeriksa apakah pengguna memiliki *work entry “shift”* atau tidak. Jika iya, maka hanya fitur *attendance system* yang akan ditampilkan, sedangkan jika tidak, akan terdapat alur tambahan untuk *lunch system*.



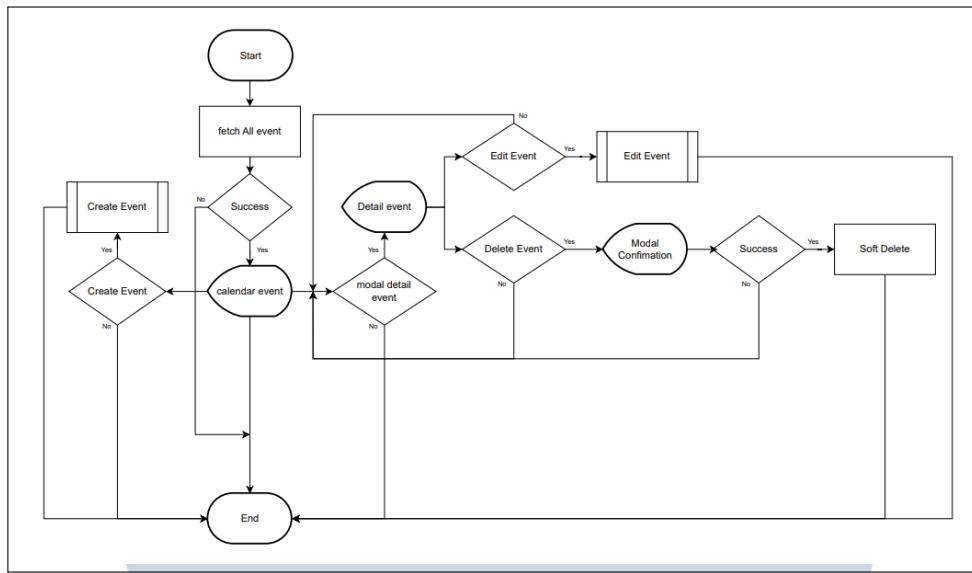
Gambar 3.11. *Flowchart Attendance System Fixed Type*

Gambar 3.11 memperlihatkan *flowchart* pada sistem *attendance* dengan jenis *work entry fixed*. Proses dimulai dengan pengecekan apakah pengguna sudah melakukan *clock-in*. Jika belum, sistem akan memeriksa keterlambatan. Apabila pengguna terlambat, mereka wajib mengisi alasannya sebelum data disimpan ke dalam *database*. Setelah *clock-in*, sistem akan menampilkan fitur *lunch system* dan memeriksa apakah pengguna sudah melakukan *lunch in*. Jika belum, pengguna dapat melakukan *lunch in*, kemudian *lunch end*. Apabila *lunch end* dilakukan terlambat, pengguna juga harus mengisi alasannya. Jika pengguna sudah melakukan *clock-in* dan *lunch in*, sistem akan memeriksa apakah pengguna mencoba *clock-out* terlalu awal maka harus mengisi alasannya. Lalu, sistem akan menyimpan data ke dalam *database*.



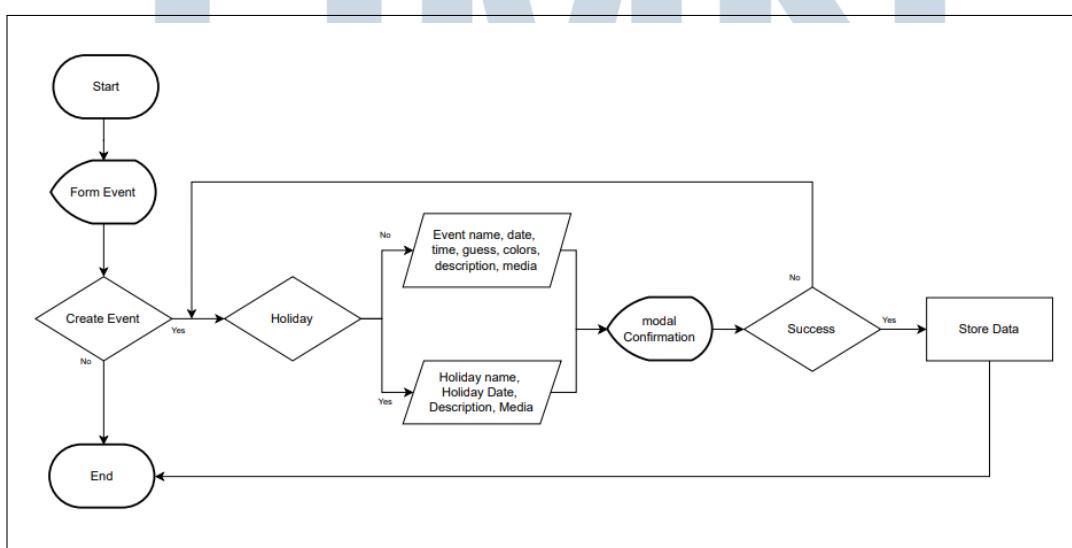
Gambar 3.12. *Flowchart Attendance System Shift Type*

Gambar 3.12 memperlihatkan *flowchart* pada sistem *attendance* dengan jenis *work entry shift*. Proses dimulai dengan pengecekan apakah pengguna sudah melakukan *clock-in*. Jika belum, sistem akan meminta jadwal *shift* dan memeriksa keterlambatan. Apabila pengguna terlambat, mereka wajib mengisi alasan keterlambatan sebelum data disimpan ke dalam *database*. Sebaliknya, jika pengguna sudah melakukan *clock-in*, sistem akan memeriksa apakah pengguna mencoba *clock-out* lebih awal. Jika iya, pengguna diwajibkan mengisi alasan sebelum data disimpan ke dalam *database*.



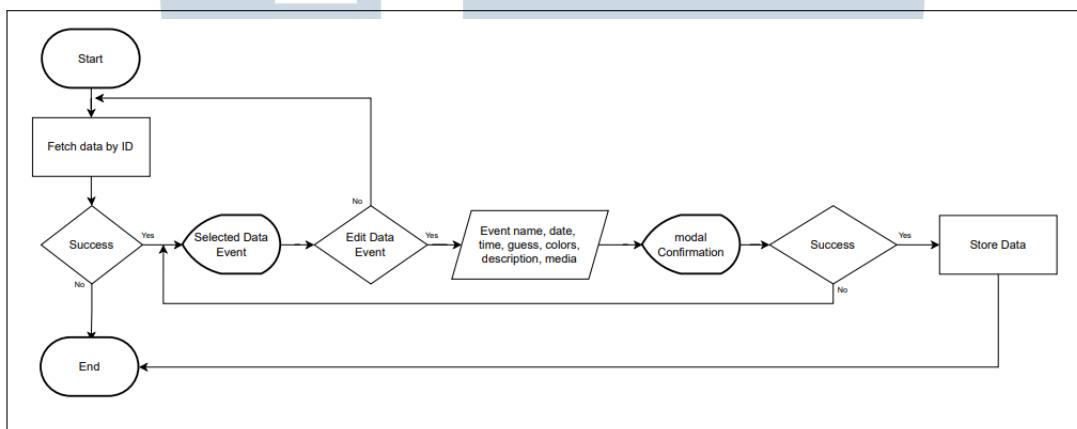
Gambar 3.13. *Flowchart Event Calendar*

Gambar 3.13 menunjukkan *flowchart* pada halaman *Event Calendar*. Proses diawali dengan pemanggilan GET API untuk mengambil data *event* dan hari libur yang telah tersimpan. Pengguna dapat melihat detail *event* melalui modal yang menampilkan informasi lengkap beserta opsi untuk mengedit atau menghapus *event*. Jika pengguna memilih untuk menghapus, sistem akan menampilkan modal konfirmasi terlebih dahulu, dan hanya pembuat *event* yang memiliki hak untuk melakukan penghapusan. Selain itu, pengguna yang memiliki *permission create* dapat membuat *event* baru.



Gambar 3.14. *Flowchart Event Create*

Gambar 3.14 memperlihatkan *flowchart* pada halaman *Create Event*. Proses dimulai dengan menampilkan *form field* untuk mengisi data *event*. Terdapat pilihan untuk *event type*, seperti *meeting* dan *holiday*. Apabila pengguna memilih *meeting*, maka data yang perlu diisi meliputi nama *meeting*, tanggal, waktu pelaksanaan, daftar karyawan yang terlibat, warna yang akan ditampilkan pada kalender, deskripsi, dan dokumen pendukung. Sementara itu, jika memilih *holiday*, pengguna hanya perlu mengisi tanggal, judul hari libur, serta *optional field* berupa deskripsi dan dokumen pendukung. Setelah pengisian data selesai, akan ditampilkan modal konfirmasi. Apabila disetujui, data *event* tersebut akan dikirimkan dan disimpan ke dalam *database*.



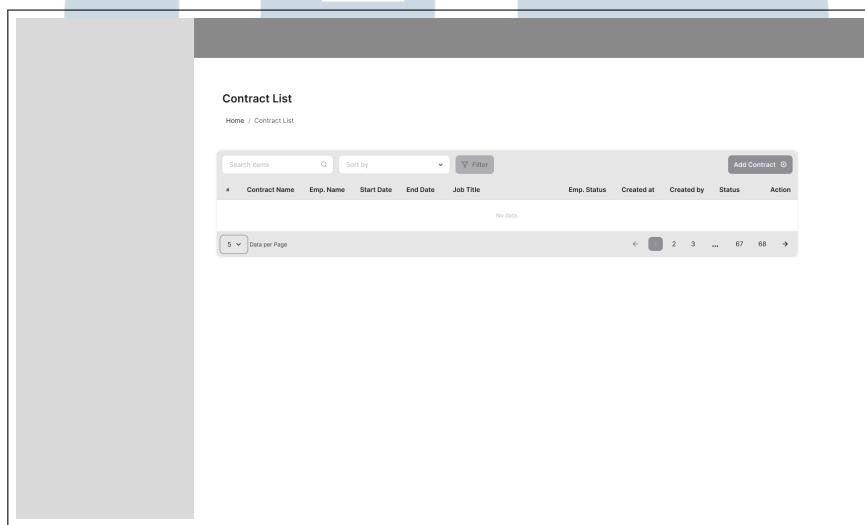
Gambar 3.15. *Flowchart Event Edit*

Gambar 3.15 memperlihatkan *flowchart* pada halaman *Edit Event*. Proses dimulai dengan pemanggilan GET API untuk mengambil data *event* yang dipilih untuk diubah. Pengguna dapat melakukan perubahan pada nama *event*, waktu pelaksanaan, daftar karyawan yang terlibat, warna *event*, deskripsi, dan dokumen pendukung. Setelah seluruh perubahan selesai dilakukan, akan ditampilkan modal konfirmasi. Apabila konfirmasi disetujui, data *event* yang telah diperbarui akan dikirimkan dan disimpan ke dalam *database*.

MULTIMEDIA
UNIVERSITAS
NUSANTARA

3.5 Wireframe

Bagian ini menampilkan *wireframe* sebagai rancangan dasar antarmuka *Human Resource Internal System*. *Wireframe* ini digunakan untuk memvisualisasikan struktur halaman, susunan elemen, serta alur interaksi pengguna sebelum sistem diimplementasikan secara penuh. Dengan adanya *wireframe*, proses perancangan dan pengembangan dapat dilakukan secara lebih terarah serta memastikan bahwa kebutuhan fungsional telah dipetakan dengan baik sejak tahap awal. Rancangan *wireframe* tersebut disajikan sebagai berikut:



Gambar 3.16. *Wireframe Contract List*

Gambar 3.16 adalah *wireframe* pada halaman *Contract List*. Pada halaman ini menampilkan tabel yang berisi seluruh daftar kontrak yang telah dibuat. Beberapa kolom yang ditampilkan meliputi *Contract Name*, *Employee Name*, *Start Date*, *End Date*, *Job Title*, *Employee Status*, *Created At*, *Created By*, *Status*, dan *Action*. Selain itu, tersedia tombol untuk membuat kontrak baru yang akan mengarahkan pengguna ke halaman *Contract Input*. Pada bagian *Action*, terdapat ikon detail (*eye*) yang digunakan untuk memperbarui status atau melihat informasi lengkap kontrak, serta ikon *edit* (*pencil*) yang memungkinkan pengubahan data kontrak selama status kontrak masih *pending*.

The wireframe for the 'Add Contract' page is organized into several sections:

- General Information:** Contains fields for Employee Name (dropdown), Contract Name (text input), Job Title (dropdown), Employment Status (dropdown), Supervisor (dropdown), Work Location (dropdown), Start Date (date input), End Date (date input), and Leave (Optional) (dropdown).
- Work Schedule:** Contains a Work Entry dropdown.
- Attendance & Penalty Rule:** Contains a checkbox for 'MIA (Missing In Action)' (Absent without notice or approval) and an amount field (Rp 0).

Gambar 3.17. Wireframe Contract Input

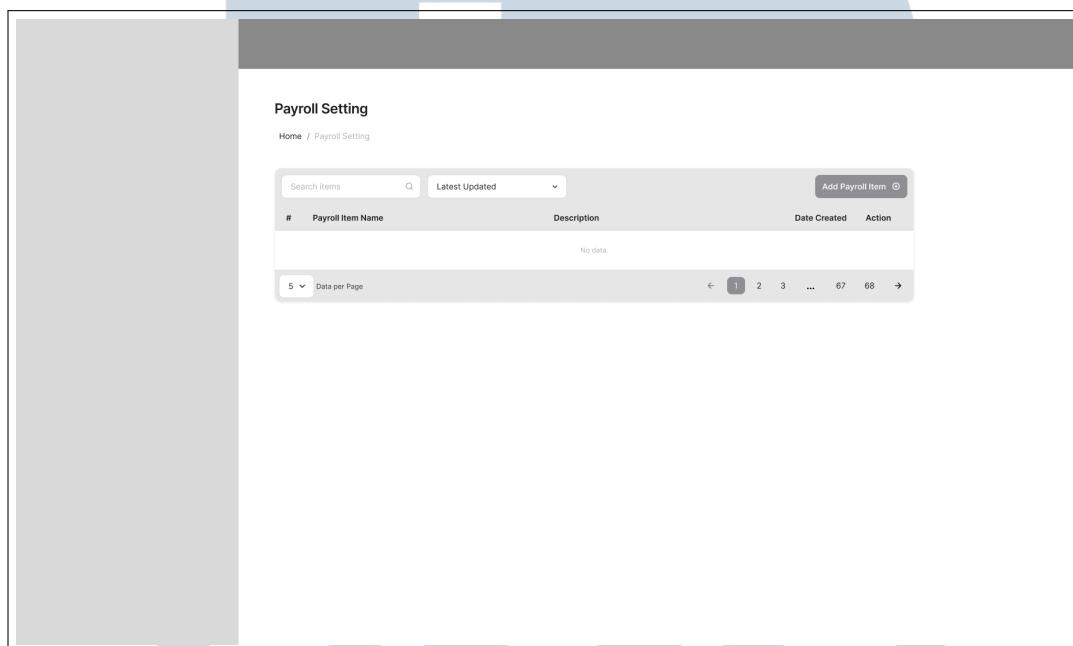
Gambar 3.17 adalah *wireframe* pada halaman *Contract Input*. Halaman ini digunakan untuk membuat kontrak baru bagi karyawan. Terdapat beberapa data yang harus dipenuhi yaitu pada bagian *General Information* terdapat *form fields* *employee name*, *contract name*, *job title*, *employee status*, *supervisor*, *work location*, *start date*, *end date*, dan *leave*. Bagian *Work Schedule* memuat *form field* untuk *work entry* dan *work schedule*. Selanjutnya, pada bagian *Attendance and Penalty Rule*, terdapat pengaturan *Missing In Action* serta berbagai penalti lain yang dapat disesuaikan dengan kebijakan perusahaan. Terakhir, bagian *Salary and Benefit* berisi informasi mengenai gaji pokok serta tambahan tunjangan atau manfaat lainnya.

The wireframe for the 'Contract Review' page displays the following details for a specific contract:

- General Information:** Employee Name: Bryant Babas, Contract Name: Bryant Contract, Job Title: Pemelihara Gedung, Employment Status: Fulltime, Supervisor: Rubyant Evelyn, Work Location: Semarang, Start Date: 16/06/2025, End Date: 16/06/2026, Leave: Annual Leave, Sick Leave, Paternity Leave.
- Work Schedule:** Work Entry: Fixed Hour, Work Days: Senin - Jumat (08:00 - 17:00), Sabtu (08:00 - 15:00), Lunch Time (12:00 - 13:00).
- Attendance & Penalty Rule:** Late Arrival (Less than 2 hours) and Penalty applied for minor tardiness below 2 hours, Amount: 11.000.

Gambar 3.18. Wireframe Contract Detail

Gambar 3.18 adalah *wireframe* pada halaman *Contract Detail*. Halaman ini menampilkan seluruh informasi kontrak yang telah dibuat dan menyediakan sejumlah aksi yang dapat dilakukan, seperti *approve*, *reject*, *cancel*, atau *terminate*, sesuai dengan status kontrak yang sedang berjalan. Apabila status kontrak adalah *pending* maka aksi yang dapat dilakukan *accept* atau *reject*. Jika status kontrak adalah *approve* maka aksi yang dapat dilakukan adalah *cancel* dan status kontrak *on going* maka aksi yang dapat dilakukan adalah *terminate*.



Gambar 3.19. *Wireframe Payroll setting List*

Gambar 3.19 adalah *wireframe* pada halaman *Payroll setting List*. Halaman ini menampilkan seluruh daftar *item payroll* yang telah dibuat dan akan digunakan pada bagian *Salary and Benefit* saat proses pembuatan kontrak. Beberapa kolom yang ditampilkan meliputi *Item Name*, *Description*, *Date Created*, dan *Action*. Selain itu, tersedia tombol untuk menambahkan *item payroll* baru yang akan mengarahkan pengguna ke halaman *Payroll Setting Input*.

The wireframe shows a form titled 'Add Payroll Item'. At the top right is a button labeled 'Add Row'. Below the title, the URL 'Home / Payroll Setting / Add Payroll Item' is displayed. The main area is divided into sections: 'Info' on the left and 'Payroll Item Name *' and 'Description' on the right. The 'Payroll Item Name' field contains 'Allowance Name'. The 'Description' field contains 'Input Description'. At the bottom right are 'Cancel' and 'Save' buttons.

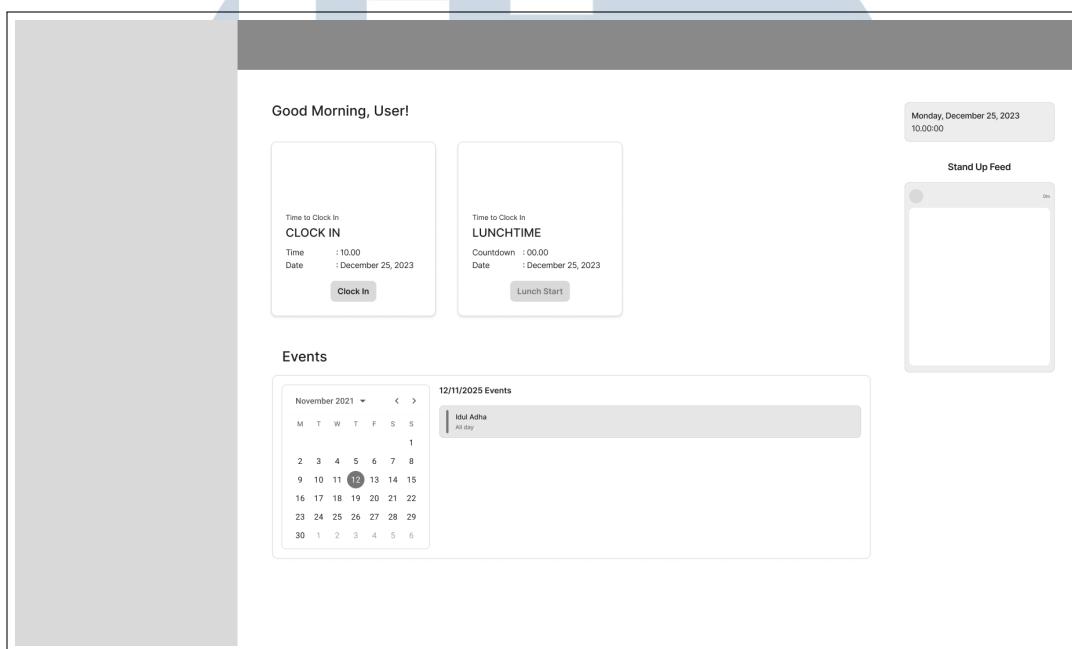
Gambar 3.20. *Wireframe Payroll setting Input*

Gambar 3.20 adalah *wireframe* pada halaman *Payroll setting Input*. Halaman ini digunakan untuk menambahkan *item payroll* baru, dan memungkinkan penambahan lebih dari satu data sekaligus melalui tombol *Add Row*. Terdapat *form fields* yang perlu diisi, yaitu *item name* dan *description*, sebelum item tersebut dapat disimpan ke dalam daftar *payroll*.

The wireframe shows a form titled 'Detail Payroll Item'. At the top right is a button labeled 'Add Row'. Below the title, the URL 'Home / Payroll Setting / Detail Payroll Item' is displayed. The main area is divided into sections: 'Info' on the left and 'Payroll Item Name' and 'Description' on the right. The 'Payroll Item Name' field contains 'Basic Salary'. The 'Description' field contains 'Based on employment contract'.

Gambar 3.21. *Wireframe Payroll setting Detail*

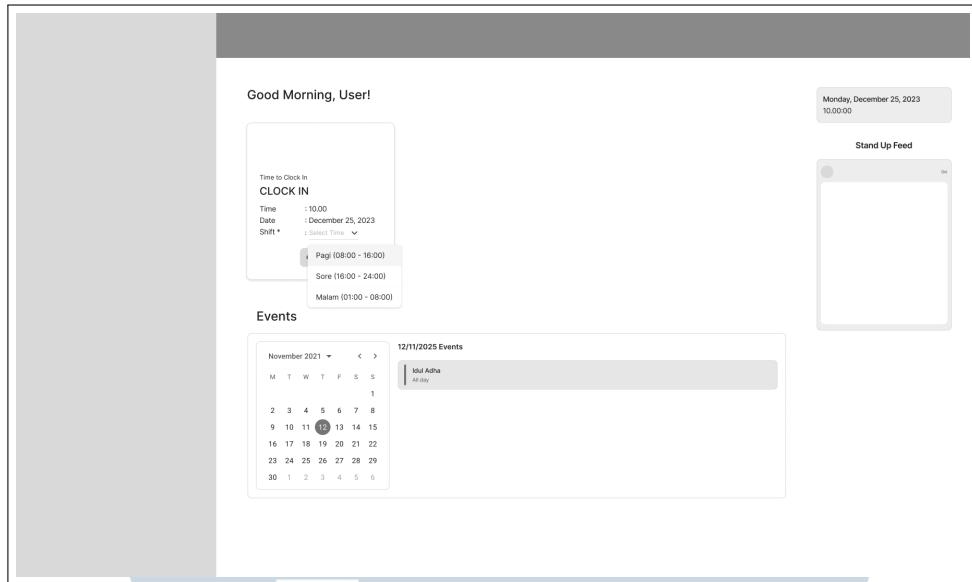
Gambar 3.21 adalah *wireframe* pada halaman *Payroll setting Detail*. Halaman ini menampilkan informasi lengkap mengenai *item payroll* yang telah dibuat, seperti *item name* dan *description*. Selain menampilkan informasi, halaman ini dapat untuk memastikan bahwa setiap komponen gaji yang didefinisikan sudah memiliki penjelasan yang cukup.



Gambar 3.21. Wireframe Payroll setting Detail

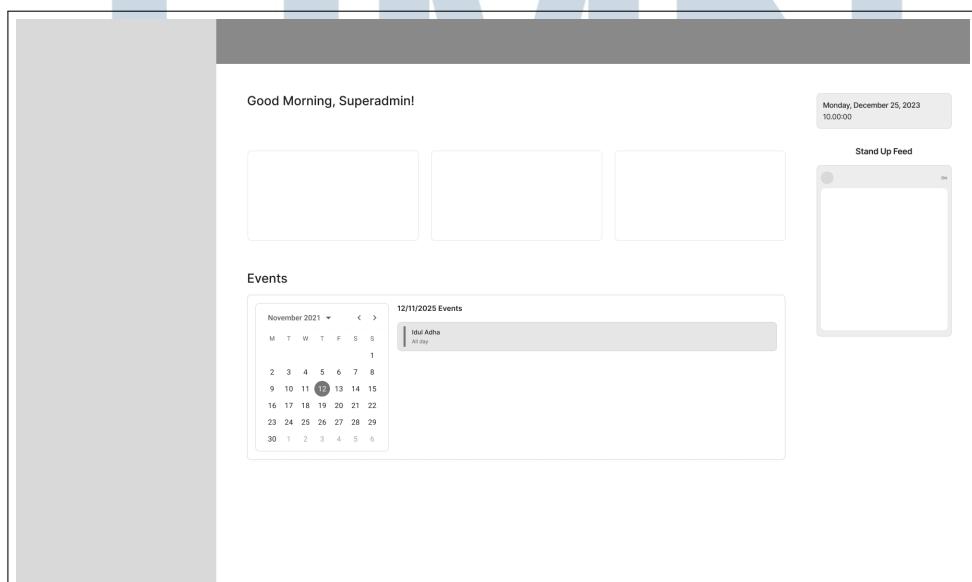
Gambar 3.22 adalah *wireframe* pada halaman *Dashboard attendance work type fixed*. Pada halaman ini, pengguna dapat melakukan proses absensi yang meliputi *clock in*, *clock out*, serta absensi untuk jam makan siang sesuai dengan jadwal yang telah ditetapkan pada kontrak kerja dengan tipe *fixed*. Apabila pengguna tidak mendapatkan jam makan siang, maka fitur tersebut tidak akan ditampilkan pada halaman ini.

Pengguna memulai proses absensi dengan melakukan *clock in* sesuai jadwal yang telah ditetapkan. Setelah itu, pengguna mengisi aktivitas kerja yang akan ditampilkan pada bagian *standup feed*. Selanjutnya, apabila pengguna memiliki jadwal makan siang, sistem *lunch* akan aktif dan menjalankan *countdown* hingga waktu makan siang berakhir. Namun, bagi pengguna yang tidak memiliki jadwal makan siang, proses absensi tetap dapat dilanjutkan dengan melakukan *clock out* sesuai dengan ketentuan yang tercantum pada kontrak kerja.



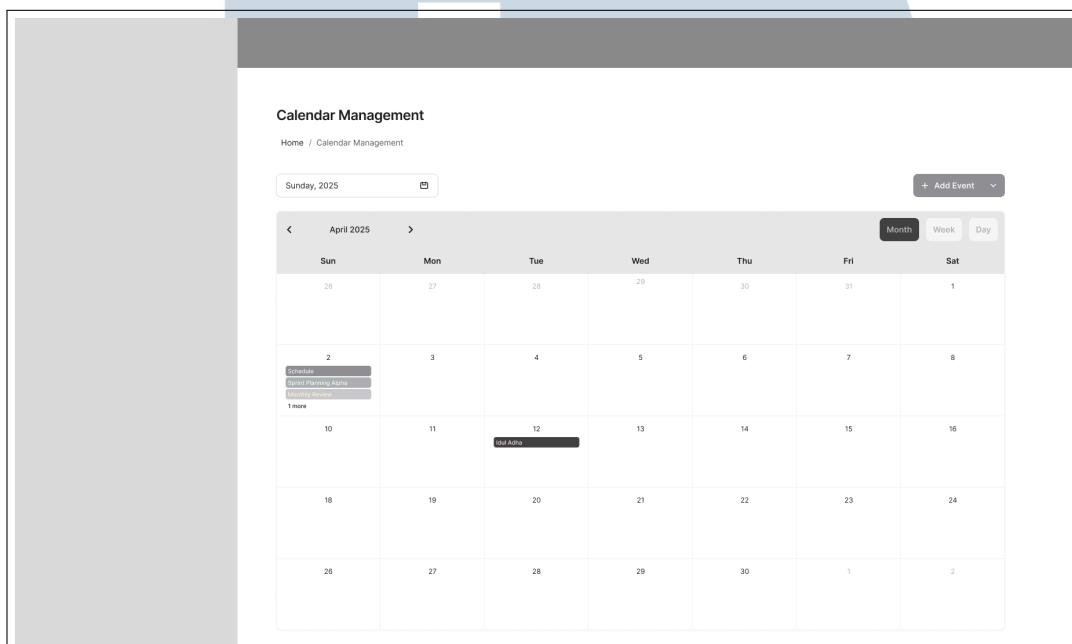
Gambar 3.23. Wireframe Attendance work type Shift

Gambar 3.23 adalah *wireframe* pada halaman *Dashboard Attendance Work Type Shift*. Pada halaman ini, pengguna dapat melakukan proses absensi dengan memilih jadwal *shift* yang akan dijalani terlebih dahulu. Pilihan *shift* tersebut ditampilkan berdasarkan jadwal kerja yang telah ditetapkan pada kontrak kerja dengan tipe *shift*. Melalui tampilan ini, sistem memastikan bahwa proses absensi dilakukan sesuai dengan jadwal yang berlaku sehingga data kehadiran karyawan dapat tercatat secara akurat.



Gambar 3.24. Wireframe Event Dashboard

Gambar 3.24 adalah *wireframe* pada halaman *Dashboard*. Pada halaman ini menampilkan daftar *event* yang tersedia sebagai *announcement* pada halaman utama, serta memungkinkan pengguna untuk melihat *event* berdasarkan tanggal yang dipilih. Detail informasi *event* dapat diakses melalui list yang tersedia dengan mengklik judul *event* tersebut. Fitur *event* ini tersedia di semua halaman *dashboard* pengguna yang memiliki akses kedalam *Human Resource Information System (HRIS)*.



Gambar 3.25. *Wireframe Event Calendar*

Gambar 3.25 adalah *wireframe* pada halaman *Event Calendar*. Pada halaman ini menampilkan daftar jadwal dalam bentuk kalender dan memungkinkan pengguna untuk melihat tampilan jadwal berdasarkan mode *month*, *week*, maupun *day*, sehingga detail tanggal dan waktu kegiatan dapat dipantau dengan lebih mudah dan terstruktur. Terdapat list kegiatan di setiap tanggalnya yang menampilkan maksimal tiga kegiatan. Untuk menambahkan kegiatan baru, pengguna dapat mengklik tombol (*add event*) yang akan mengarahkan ke halaman *Event Input*.

Add Event

Home / Calendar Management / Add Event

Info

Event Type * Meeting Event Name * Input Event Name Date * DD/MM/YYYY - DD/MM/YYYY

Start Time * 00:00 WIB End Time * 01:00 WIB Color Select Color

Guest(s) * Select User(s) Description Input Description

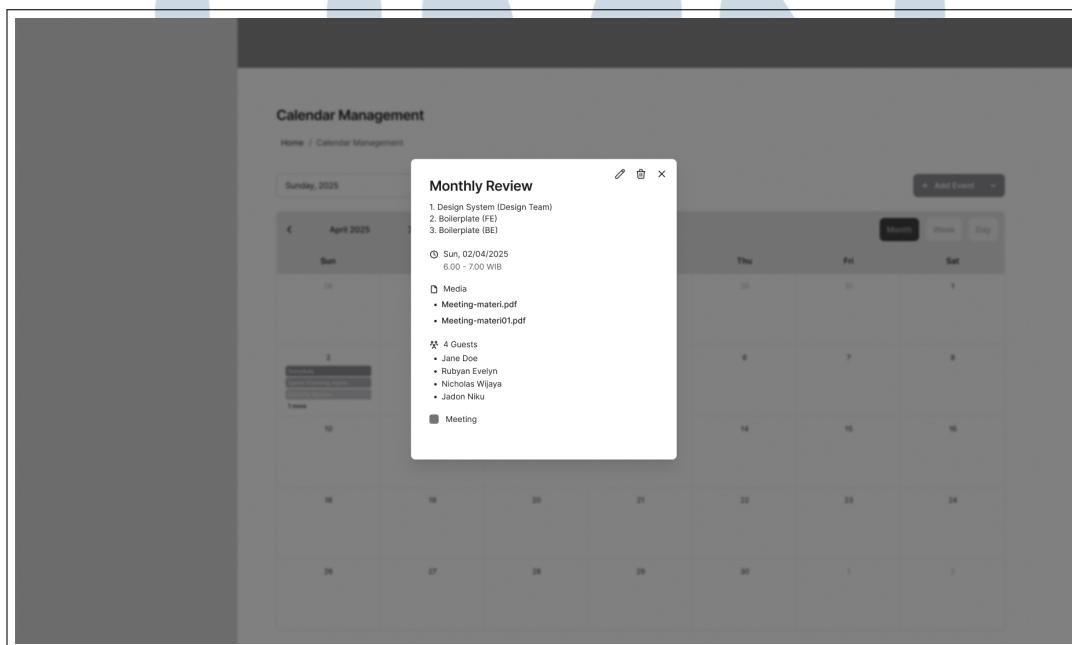
Upload Media

Upload Document Format : PDF, JPG, PNG Choose File

Cancel Save

Gambar 3.26. Wireframe Event Input

Gambar 3.26 adalah *wireframe* pada halaman *Event Input*. Pada halaman ini digunakan untuk menambahkan *event* baru dengan memilih jenis *event* terlebih dahulu, seperti *meeting*, *holiday*, atau *schedule*. Setelah menentukan jenisnya, pengguna dapat mengisi sejumlah *form field* yang tersedia, antara lain *event name*, *date*, *start time*, *end time*, *color*, *guest*, *description*, serta mengunggah dokumen pendukung sebagai kelengkapan informasi terkait kegiatan yang akan dilaksanakan.



Gambar 3.27. Wireframe Event Modal

Gambar 3.27 adalah *wireframe* pada bagian *Event Modal* pada halaman *Event Calendar*. Bagian ini menampilkan detail lengkap dari jadwal yang dipilih, meliputi judul kegiatan, deskripsi, media atau *file* terkait, daftar tamu yang diundang, serta tipe jadwal yang sedang ditampilkan. File tersebut dapat diakses langsung melalui tautan yang tersedia. Selain itu, terdapat opsi untuk mengedit atau menghapus jadwal, di mana hanya pembuat jadwal yang memiliki hak akses untuk melakukan penghapusan.

3.6 Implementasi

Human Resource Internal System telah berhasil dikembangkan dan diimplementasikan pada mode *production*, serta resmi dijadikan produk yang digunakan oleh klien terkait. Berikut ini merupakan tampilan akhir dari *Human Resource Internal System* yang telah selesai dikembangkan:

#	Contract Name	Emp. Name	Start Date	End Date	Job Title	Emp. Status	Status	Actions
1	Contract Jose	Jose	10/10/2025	26/01/2026	Software Engineer	Internship	Approved	
2	aced	Amanda Citra	05/10/2025	31/10/2025	halo	Probation	Approved	
3	kej	Keshia	14/10/2025	28/10/2025	Bukan Admin	Full Time	Pending	
4	bARU	userbaru	02/10/2025	30/10/2025	makan	Probation	Pending	
5	Trash Contract	Trash	02/10/2025	03/10/2025	hmm	Internship	Cancelled	
6	KC	Galista	26/09/2025	30/09/2025	DevOps Engineer	Full Time	Terminated	
7	KC	Kevin Ken	26/09/2025	30/09/2025	Backend Engineer	Full Time	Terminated	
8	JC	Jose	26/09/2025	30/09/2025	Backend Engineer	Internship	Pending	
9	KC	Keshia	26/09/2025	30/09/2025	Backend Engineer	Full Time	Terminated	

Gambar 3.28. Tampilan *Contract List*

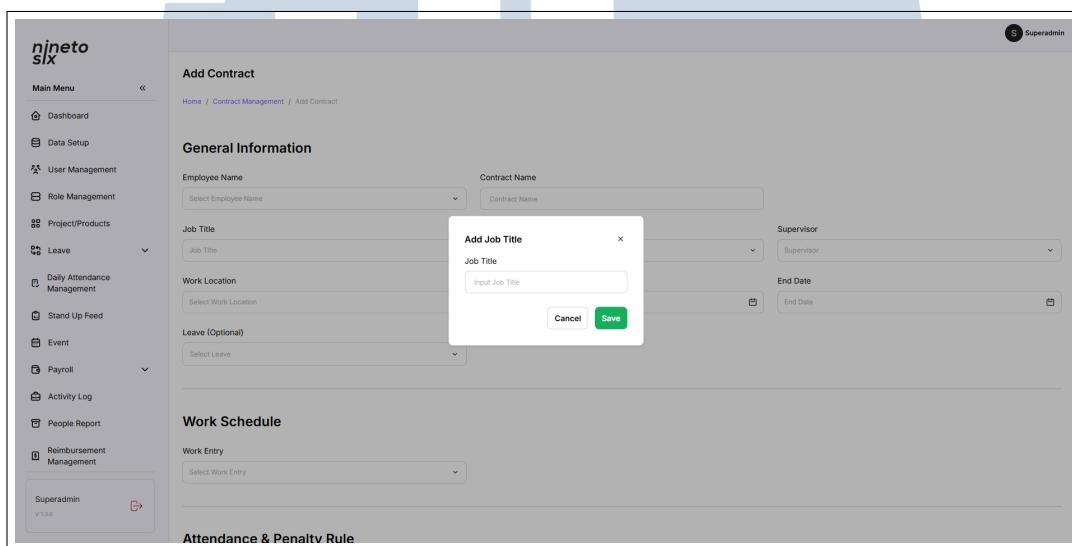
Gambar 3.28 menunjukkan tampilan halaman *Contract List*. Pada halaman ini terdapat tabel yang menampilkan daftar kontrak yang telah dibuat. Beberapa kolom yang ditampilkan meliputi *Contract Name*, *Employee Name*, *Start Date*, *End Date*, *Job Title*, *Employee Status*, *Created At*, *Created By*, *Status*, dan *Action*. Jika *user* ingin membuat kontrak baru, tersedia tombol yang mengarahkan ke halaman *Contract Input*. Untuk kontrak yang berstatus pending, terdapat aksi *Edit* (ikon *pencil*) yang membawa *user* ke halaman *edit* kontrak untuk melakukan pembaruan atau penghapusan data. Selain itu, tersedia ikon detail (*eye*) yang digunakan untuk memperbarui status kontrak atau melihat informasi lengkap dari kontrak tersebut.

Gambar 3.29. Tampilan *Contract Input General Information* dan *Work Schedule*

Gambar 3.29 menunjukkan tampilan halaman *Contract Input*. Pada halaman ini, user dapat membuat kontrak baru dengan mengisi sejumlah *form field* yang telah disediakan. Pada bagian *General Information*, terdapat *form field* seperti *employee name*, *contract name*, *job title*, *employee status*, *supervisor*, *work location*, *start date*, *end date*, dan *leave*. Sementara itu, bagian *Work Schedule* mencakup *form field* untuk *work entry* dan *work schedule*, di mana opsi pada *work schedule* akan menyesuaikan secara otomatis berdasarkan pilihan *work entry* yang dipilih, yaitu antara tipe *fixed* atau *shift*.

Gambar 3.30. Tampilan *Contract Input Attendance and Penalty Rule* dan *Salary and Benefit*

Gambar 3.30 menunjukkan tampilan halaman *Contract Input*. Pada halaman ini, user dapat melanjutkan dalam pembuatan kontrak baru dengan mengisi sejumlah *form field* yang telah disediakan. Pada bagian *Attendance and Penalty Rule*, terdapat pengaturan mengenai *Missing In Action* yang wajib di isi serta berbagai jenis penalti lain yang dapat disesuaikan dengan kebijakan perusahaan. Terakhir, bagian *Salary and Benefit* memuat informasi terkait gaji pokok yang diharuskan di isi dengan *unit addition* serta tambahan tunjangan atau manfaat lainnya.



Gambar 3.31. Tampilan Fitur *Add Job Title* pada *Contract Input*

Gambar 3.31 menunjukkan tampilan fitur *Add Job Title* pada halaman *Contract Input*. Fitur ini memungkinkan *user* untuk menambahkan *job title* baru secara langsung tanpa harus meninggalkan halaman pembuatan kontrak. Dengan mengklik ikon tambah (*plus*) di dalam *option form field job title*, akan muncul sebuah modal yang berisi *form field* untuk mengisi nama *job title* baru. Setelah data diisi, pengguna dapat menyimpan perubahan tersebut, dan *job title* baru akan langsung tersedia untuk dipilih pada halaman *Contract Input*.

Gambar 3.32. Tampilan Fitur *Leave* pada *Contract Input*

Gambar 3.32 menunjukkan tampilan fitur *Leave* pada halaman *Contract Input*. Fitur ini berfungsi untuk menampilkan jumlah cuti yang berlaku bagi karyawan sesuai ketentuan yang telah ditetapkan pada *Leave Setting*. Sistem akan melakukan penyaringan (*filter*) secara otomatis berdasarkan *job title* dan *employee status*, sehingga hak cuti yang ditampilkan sudah sesuai dengan kebijakan perusahaan untuk karyawan tersebut.

Gambar 3.33. Tampilan *Leave Setting Input*

Gambar 3.33 menunjukkan tampilan halaman *Leave Setting Input*. Pada halaman ini, *user* dapat membuat pengaturan cuti baru yang nantinya akan digunakan pada bagian *Leave* saat proses pembuatan kontrak. Tersedia beberapa *form field* yang dapat diisi, yaitu *Leave Name*, *Job Title*, *Employee Status*, *Leave Duration*, dan *Leave Interval*. Pengaturan cuti ini dapat disesuaikan (*customized*) berdasarkan *job title*, *employee status*, atau kombinasi keduanya, sehingga hak cuti dapat diatur secara lebih spesifik sesuai kebutuhan perusahaan dan dapat difilter secara otomatis ketika kontrak karyawan dibuat.

The screenshot shows the 'Contract Review' section of the application. On the left is a sidebar with various menu items like Dashboard, User Management, Role Management, Project/Products, Leave, Daily Attendance Management, Stand Up Feed, Event, Payroll, Activity Log, People Report, and Reimbursement Management. A user profile for 'Superadmin' is at the bottom of the sidebar. The main content area has a header 'Contract Review' with a breadcrumb 'Home / Contract Management / Contract Review'. It shows the following data:

General Information		Approved	
Employee Name	Jose	Contract Name	Contract Jose
Job Title	Software Engineer	Employee Status	Internship
Work Location	Tangerang	Start Date	10/10/2025
Leave	WFH Permit	End Date	26/01/2026
<hr/>			
Work Schedule			
Work Entry	Fixed	Work Days	<ul style="list-style-type: none"> • Senin - Jumat (08:00 - 17:00) • Lunch Time (12:00 - 13:00)
<hr/>			
Attendance & Penalty Rule			

Gambar 3.34. Tampilan *Contract Detail*

Gambar 3.34 menunjukkan tampilan halaman *Contract Detail*. Pada halaman ini menampilkan informasi lengkap dari kontrak yang telah dibuat dan menyediakan beberapa opsi tindakan sesuai dengan status kontrak. Jika kontrak berstatus *pending*, tersedia pilihan *approve* dan *reject*. Apabila kontrak berstatus *approved*, pengguna dapat melakukan *cancel contract*. Sementara itu, untuk kontrak dengan status *on-going*, tersedia opsi *terminate*. Jika kontrak berstatus *rejected*, maka kontrak tersebut dianggap tidak berlaku dan harus dibuat ulang melalui pembuatan kontrak baru.

The screenshot shows the 'Payroll Setting' page in the nineto SIX application. The left sidebar has a 'Payroll' section with 'Payroll Setting' selected. The main area displays a table titled 'Payroll Setting' with the following data:

#	Item Name	Description	Date Created	Actions
1	Tunjangan BRUS	Asumsi Kesehatan	17/09/2023	
2	Tunjangan Transportasi	Jaminan Transportasi	17/09/2023	
3	Tunjangan Makan	Uang Jajan	17/09/2023	
4	Main Salary	Gaji Pokok	17/09/2023	-
5	RPH	Pajak	30/08/2023	

Gambar 3.35. Tampilan *Payroll setting List*

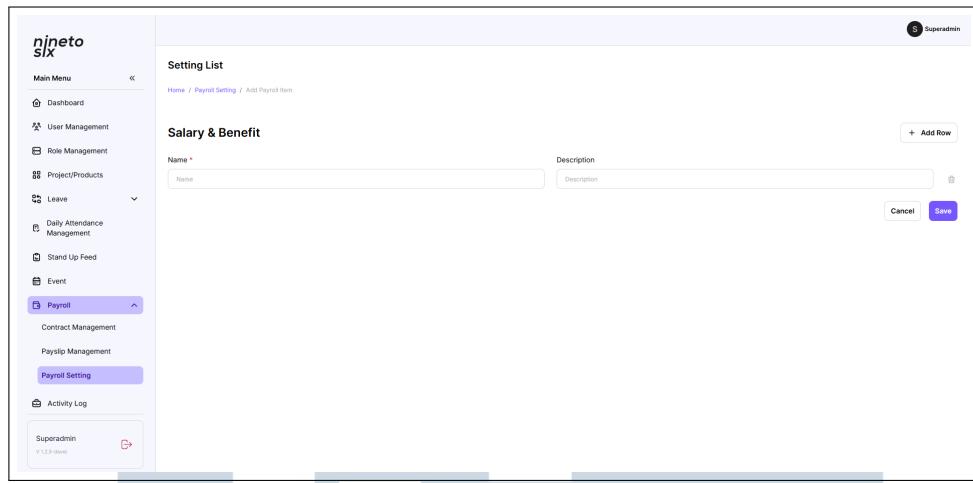
Gambar 3.35 menunjukkan tampilan halaman *Payroll Setting List*. Halaman ini menampilkan daftar *item payroll* yang telah dibuat, dengan beberapa kolom seperti *Item Name*, *Description*, *Date Created*, dan *Action*. Tersedia tombol *Add Payroll Item* untuk menambahkan *item payroll* baru. Selain itu, pada setiap item terdapat aksi *Edit* (ikon *pencil*) yang akan mengarahkan pengguna ke halaman *Payroll Edit*, serta aksi *Delete* (ikon tempat sampah). Namun, untuk item Gaji Pokok/*Main Salary*, seluruh aksi tersebut tidak dapat digunakan karena item tersebut bersifat wajib (*required*) dan tidak dapat diubah maupun dihapus.

The screenshot shows the 'Payroll Setting' page with a modal dialog titled 'Delete Payroll Item' overlaid on the table. The modal contains the message 'Are you sure you want to delete from system?' with 'Cancel' and 'Delete' buttons. The table data is as follows:

#	Item Name	Description	Date Created	Actions
1	Tunjangan Dinas Keluar Kota	Bonus Dinas Keluar Kota	01/10/2023	
2	Tunjangan Kehormatan	Tunjangan yang :	21/10/2023	
3	Tunjangan Makan	Tunjangan yang :	21/10/2023	
4	Tunjangan BRUS	Tunjangan yang :	21/10/2023	
5	Gaji Pokok	Komponen utama penghasilan karyawan yang menjadi dasar perhitungan tunjangan dan potongan.	21/10/2023	-

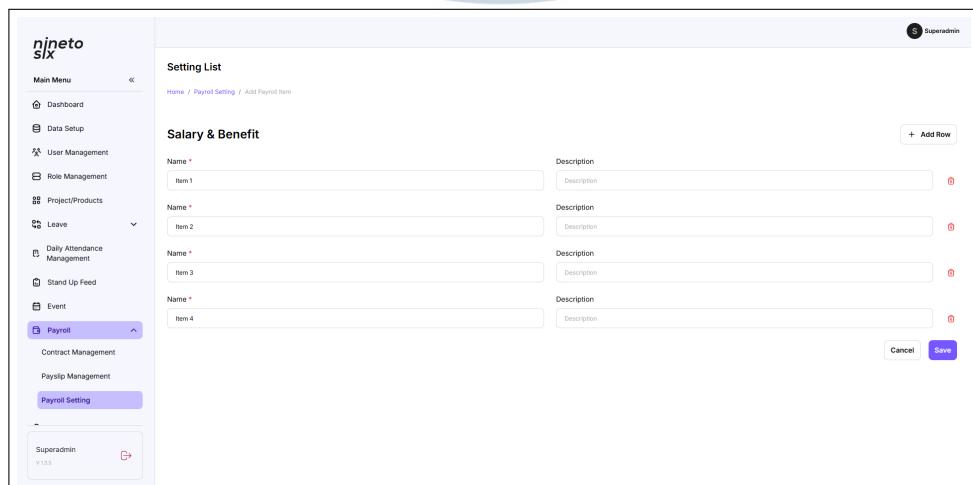
Gambar 3.36. Tampilan *Payroll setting List* Modal Delete

Gambar 3.36 menunjukkan tampilan modal *Delete* pada halaman *Payroll Setting List*. Modal ini berfungsi sebagai konfirmasi sebelum suatu item *payroll* dihapus dari sistem. Dengan adanya modal ini, pengguna dapat memastikan tindakan penghapusan benar-benar diinginkan sehingga mengurangi risiko kesalahan penghapusan data.



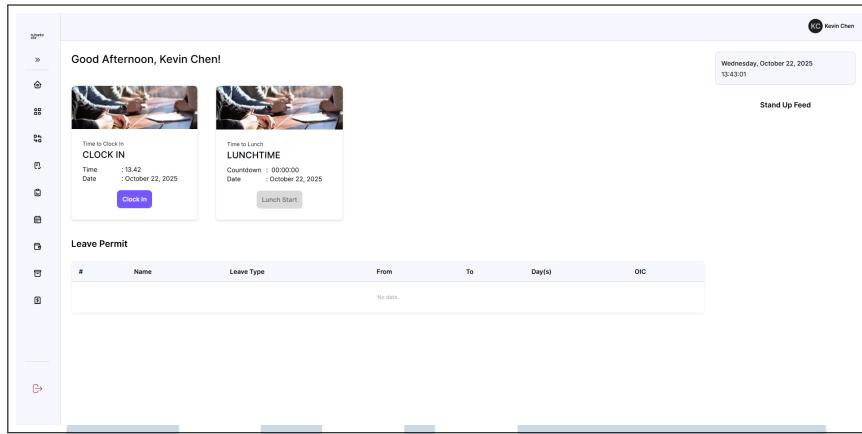
Gambar 3.37. Tampilan *Payroll Setting Input*

Gambar 3.37 menunjukkan tampilan halaman *Payroll Setting Input*. Pada halaman ini, *user* dapat membuat item *payroll* baru yang nantinya dapat dipilih saat proses pembuatan kontrak pada bagian *salary and benefit*. Tersedia beberapa *form field* utama, yaitu *Item Name* dan *Description*, yang digunakan untuk mendefinisikan detail item *payroll* yang akan ditambahkan ke dalam sistem.



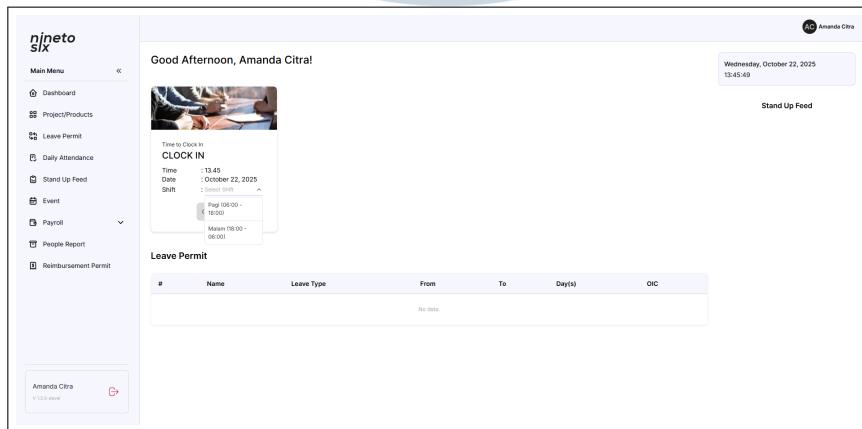
Gambar 3.38. Tampilan *Payroll Setting Input Add Row*

Gambar 3.38 menampilkan kondisi halaman *Payroll Setting Input* ketika fitur *Add Row* digunakan. Fitur ini memungkinkan *user* untuk menambahkan lebih dari satu item *payroll* dalam satu kali proses input. Dengan adanya fitur ini, proses pengelolaan data *payroll* menjadi lebih efisien karena pengguna tidak perlu melakukan penginputan item satu per satu secara terpisah.



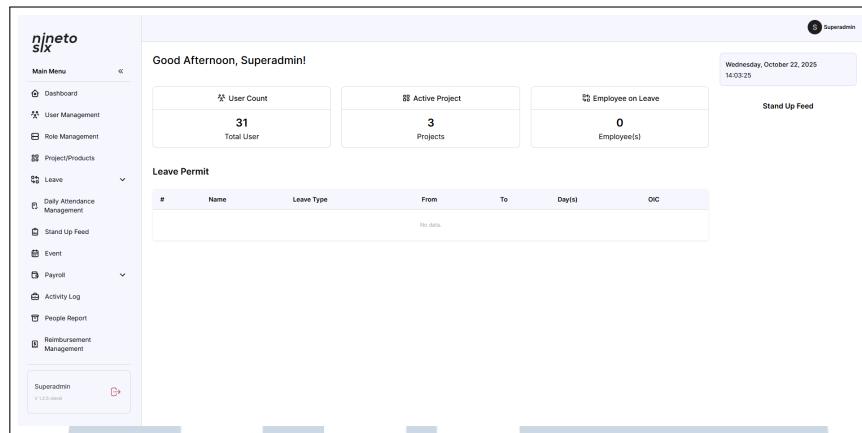
Gambar 3.39. Tampilan *Attendance work type Fixed*

Gambar 3.39 menunjukkan tampilan halaman *Dashboard*. Pada halaman ini, *user* dapat melakukan sistem *attendance* dengan *work type Fixed*, sehingga akan muncul *card* tambahan *Lunch Attendance* apabila terdapat jadwal *lunch* pada kontrak yang telah dibuat. Sistem *attendance* akan aktif berdasarkan jadwal yang ditetapkan di dalam kontrak, dan fitur *Lunch Time* akan aktif setelah *user* melakukan *Clock In*.



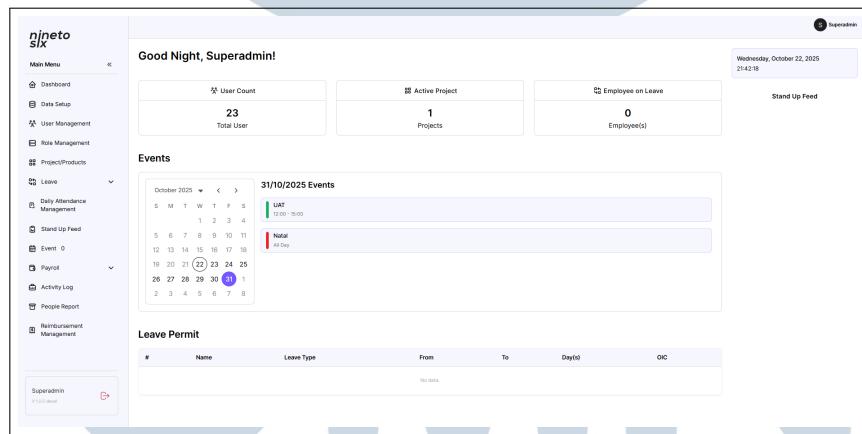
Gambar 3.40. Tampilan *Attendance work type Shift*

Gambar 3.40 menunjukkan tampilan halaman *Dashboard*. Pada halaman ini, *user* dapat melakukan *attendance* dengan *work type Shift*, sehingga tersedia *dropdown* untuk memilih jadwal *shift* sebelum melakukan *Clock In*. Sistem akan menolak proses *Clock In* apabila *user* tidak memilih *shift* yang sesuai dengan waktu masuk yang ditentukan.



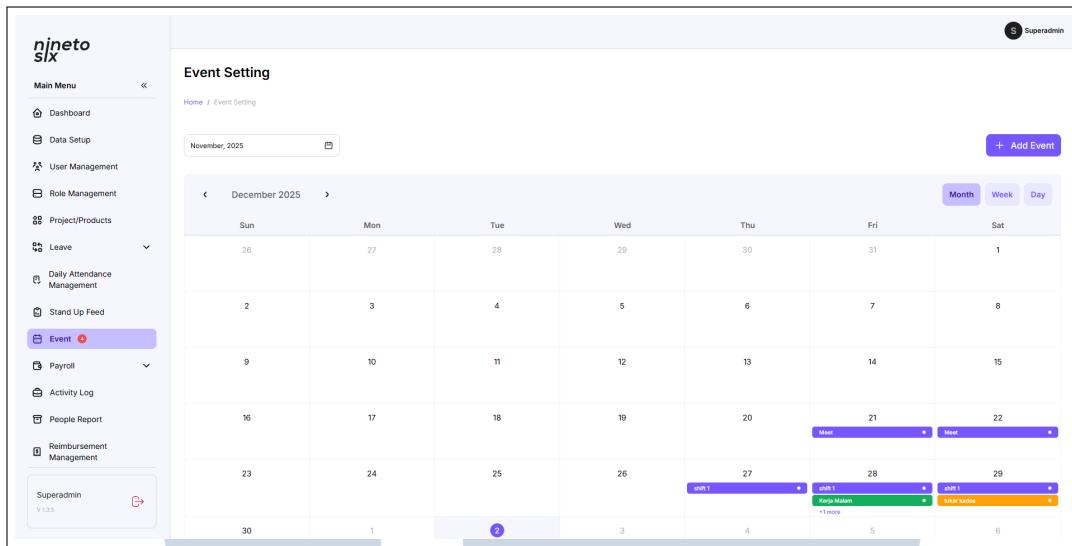
Gambar 3.41. Tampilan halaman *Dashboard Before*

Gambar 3.41 menunjukkan tampilan *Dashboard* sebelum ditambahkan fitur *Event Announcement*. Pada tampilan sebelumnya, belum tersedia informasi terkait acara yang berlangsung, sehingga menyulitkan *user* untuk mengetahui acara apa saja yang ada pada hari tertentu dan mengharuskan mereka membuka halaman *Event* terlebih dahulu.



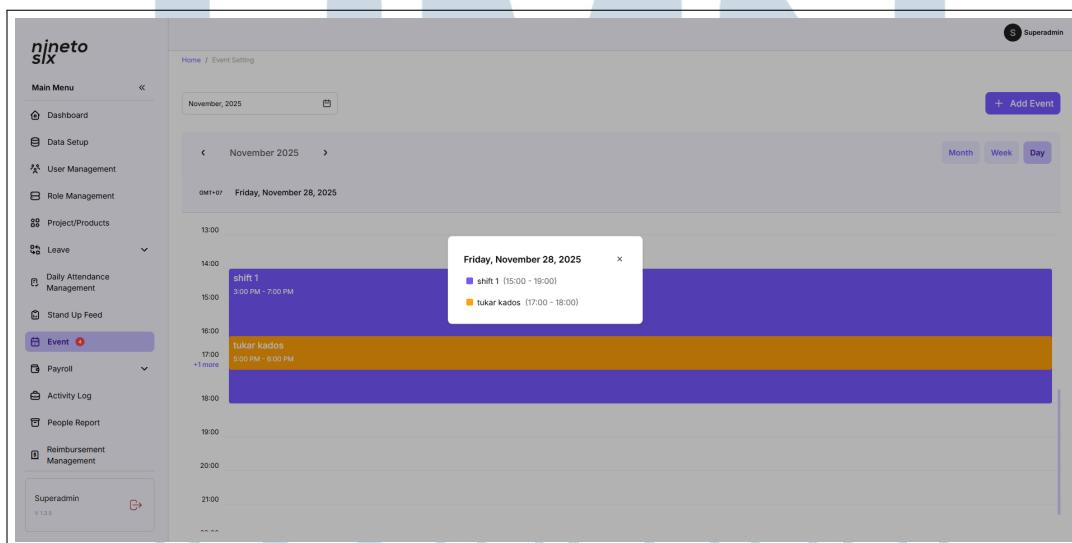
Gambar 3.42. Tampilan halaman *Dashboard After*

Gambar 3.42 menunjukkan tampilan *Dashboard* setelah penambahan fitur *Event Announcement*. Fitur ini berfungsi untuk menampilkan informasi acara yang diadakan berdasarkan hari yang dipilih, sehingga memudahkan *user* untuk mengetahui acara yang tersedia maupun yang baru ditambahkan tanpa perlu membuka halaman *Event* terlebih dahulu. Selain itu, fitur ini telah dilengkapi dengan implementasi *WebSocket* untuk memberikan pembaruan informasi acara secara *real-time*, sehingga setiap perubahan dapat langsung terlihat oleh pengguna.



Gambar 3.43. Tampilan Jadwal Event

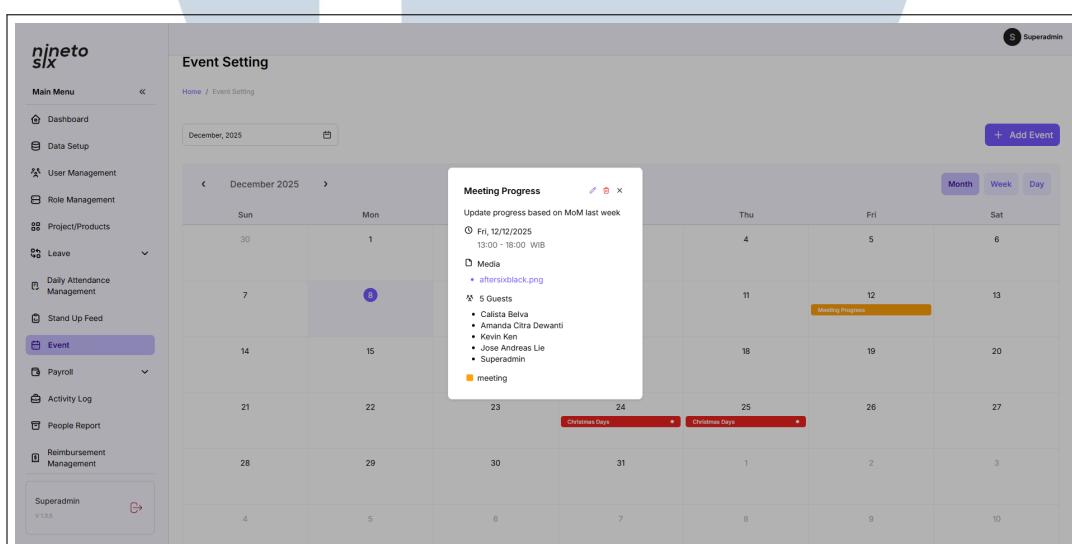
Gambar 3.43 menunjukkan tampilan halaman *Event Calendar* yang menampilkan seluruh kegiatan yang telah dibuat berdasarkan tanggal dan waktu yang telah ditetapkan. Pada halaman ini, pengguna dapat melihat daftar kegiatan secara ringkas dalam tampilan kalender. Apabila terdapat lebih dari dua kegiatan pada tanggal yang sama, sistem menyediakan tombol *See More* yang akan mengarahkan pengguna ke tampilan mode *Day* untuk melihat seluruh kegiatan pada hari tersebut secara lebih detail.



Gambar 3.44. Tampilan Jadwal Event collapse

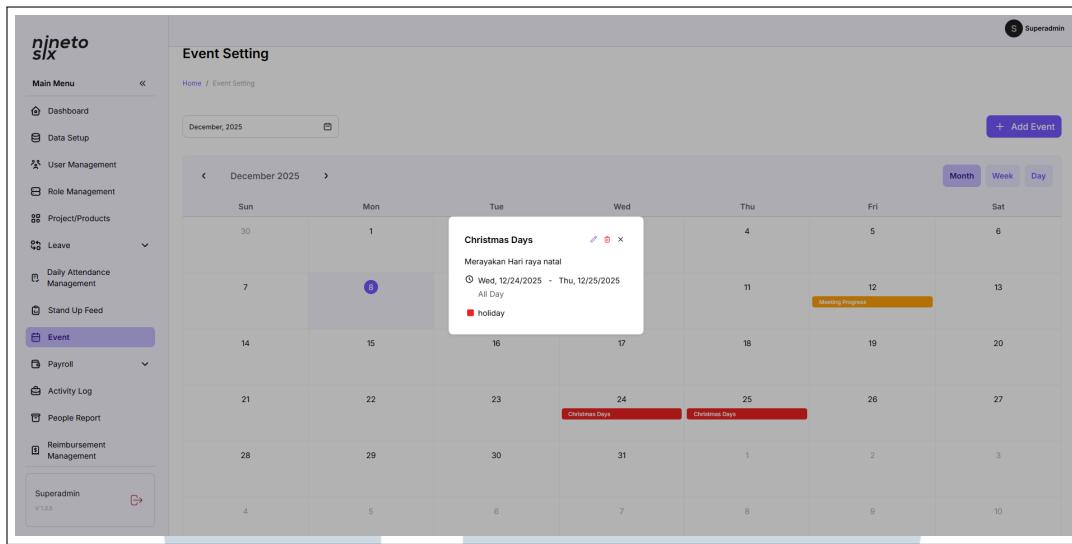
Gambar 3.44 menampilkan kondisi ketika terdapat jadwal kegiatan yang saling bertumpukan atau *collapse* pada waktu yang sama. Pada kondisi ini, sistem akan menampilkan sebuah *modal* yang berisi daftar kegiatan dengan waktu yang saling bertabrakan. Melalui tampilan tersebut, pengguna dapat mengetahui seluruh kegiatan yang berlangsung pada waktu yang sama tanpa kehilangan informasi terkait jadwal lainnya.

Selain itu, sistem juga menyediakan indikator notifikasi untuk membantu pengguna memantau kegiatan yang belum ditinjau. Pada menu *Event* di sidebar, ditampilkan *badge notification* yang menunjukkan jumlah total *event* yang belum dibuka atau dilihat detailnya. Di dalam kalender, *event* yang belum dibuka ditandai dengan sebuah titik (*dot*) berwarna putih sebagai penanda visual bahwa informasi kegiatan tersebut masih perlu diperhatikan oleh pengguna.



Gambar 3.45. Tampilan modal detail *Event type Meeting* atau *Schedule*

Gambar 3.45 menunjukkan tampilan modal detail *Event Calendar* untuk tipe *meeting* atau *schedule*. Pada bagian ini ditampilkan informasi lengkap mengenai kegiatan, seperti judul *event*, deskripsi, media atau dokumen terkait, tanggal dan waktu pelaksanaan, daftar tamu yang diundang, serta tipe *event*.



Gambar 3.46. Tampilan modal detail *Event type Holiday*

Gambar 3.46 menampilkan modal detail *Event Calendar* untuk tipe *holiday*. Modal ini hanya berisi informasi judul hari libur, deskripsi, serta tanggal pelaksanaan. Untuk tipe *holiday*, *event* secara otomatis dikirimkan ke seluruh pengguna dan diatur berlangsung sepanjang hari (*all day*).

Selain itu, terdapat tombol aksi *Edit* (ikon *pencil*) yang mengarahkan pengguna ke halaman Event Edit untuk memperbarui kegiatan, serta aksi *Delete* (ikon tempat sampah) untuk menghapus *event*. Aksi ini hanya dapat dilakukan oleh pengguna yang membuat kegiatan tersebut, sedangkan pengguna yang diundang hanya dapat melihat detail informasi tanpa dapat melakukan perubahan.

Event Name *	Event Type *	Event Date *
<input type="text" value="Event Name"/>	Select Event Type	YYYY-MM-DD to YYYY-MM-DD
Start Time *	End Time *	Color
00:00	00:00	Purple
Guest(s) *	Anyone?	
<input type="button" value="Cancel"/> <input type="button" value="Save"/>		

Gambar 3.47. Tampilan halaman *Event Input Before*

Gambar 3.47 menunjukkan tampilan halaman *Event Input* sebelum dilakukan *revamp*. Pada halaman tersebut, *user* dapat membuat *event*, namun belum tersedia fitur untuk menambahkan *supporting document* atau *description* yang diperlukan untuk memberikan detail tambahan mengenai *event* yang akan dibuat.

The screenshot shows the 'Event Setting Input' page. On the left is a sidebar with various icons and a 'Main Menu'. The main area has a header 'Event Setting Input' and a sub-header 'Home / Event Setting / Event Setting Input'. Below this is a section titled 'Info' containing fields for 'Event Type' (dropdown), 'Event Name', 'Event Date' (date range input), 'Start Time' (time input), 'End Time' (time input), 'Guest(s)' (dropdown), and 'Description' (text input). At the bottom is a section titled 'Upload Media' with a 'Upload Document' field and a 'Choose File' button. A 'Cancel' and 'Save' button are at the very bottom right.

Gambar 3.48. Tampilan halaman *Event Input After*

Gambar 3.48 menunjukkan tampilan halaman *Event Input* setelah dilakukan *revamp* pada bagian *form input*. Pada halaman ini ditambahkan *form field* *Description* untuk memberikan catatan terkait *event*, serta fitur *Upload Media* untuk mengunggah *file* yang diperlukan, seperti *file* PPT untuk keperluan *meeting*.

This screenshot shows the same 'Event Setting Input' page after the revamp. The structure is identical to the previous version, but it includes a new 'Description' field below the 'Event Name' and 'Event Date' fields. The 'Upload Media' section is also present with its own 'Description' field and file input field.

Gambar 3.49. Tampilan halaman *Event Input type Holiday*

Gambar 3.49 menunjukkan tampilan halaman *Event Input* dengan tipe *holiday*. Pada bagian *form input* untuk tipe ini, hanya terdapat *form field* *Event Name* untuk memberikan nama kegiatan libur, tanggal pelaksanaan yang ditetapkan, serta *form field opsional* seperti *description* dan upload document sebagai kelengkapan informasi terkait hari libur tersebut.

3.7 Kendala dan Solusi yang Ditemukan

Beberapa kendala yang ditemui selama melaksanakan kegiatan magang di PT Visi Karya Nusantara adalah sebagai berikut:

1. Terjadinya ketidaksesuaian komunikasi antara tim *backend* dan *frontend* pada tahap awal magang, yang menyebabkan terjadinya miskomunikasi dan memerlukan penyesuaian tambahan saat proses integrasi.
2. Kesulitan dalam menentukan alur fitur atau modul yang akan dikembangkan, sehingga membutuhkan waktu lebih untuk melakukan analisis serta riset yang mendalam.

Adapun solusi yang diterapkan untuk mengatasi kendala tersebut adalah sebagai berikut:

1. Meningkatkan kualitas komunikasi antar tim *backend* dan *frontend* guna memastikan kesesuaian saat proses integrasi.
2. Memperdalam pemahaman terhadap konsep fitur yang dikembangkan serta melakukan riset lebih lanjut terkait kebutuhan dan ekspektasi pengguna terhadap fitur atau modul tersebut.