

## BAB 3 PELAKSANAAN KERJA MAGANG

### 3.1 Kedudukan dan Organisasi

Berkedudukan sebagai *Intern Fullstack developer* pada divisi Basic Module dengan *jobdesk* untuk melakukan *debugging* terhadap fungsi yang dibuat, dan membuat serta mengembangkan fitur-fitur baru berdasarkan permintaan *client* sebagai *Backend developer/programmer*. Selain itu dilakukan juga tugas *Frontend programmer*, yaitu melakukan perubahan terhadap bagian UI (*User Interface*)/UX (*User Experience*) pada aplikasi berbasis *website*. *Database* juga merupakan salah satu *jobdesk* yang dilakukan, seperti merancang *table*, menetapkan *column* yang diperlukan, dan melakukan tuning *database* dengan meng-optimalkan ukuran data yang dipakai.

### 3.2 Tugas yang Dilakukan

Selama menjalani magang di Indotim, peran saya sebagai karyawan magang melibatkan tanggung jawab sebagai *Fullstack developer*. Tanggung jawab ini mencakup melakukan *debugging* pada fitur, mengembangkan fitur dalam aplikasi berbasis web, merancang tampilan website, serta sedikit perancangan dan pembuatan database untuk mendukung fitur tertentu. Selain itu, saya memiliki kewajiban untuk memonitor error logs dan melaporkannya ke dalam ruang obrolan (chat room) yang disediakan oleh tim pengembang perusahaan.

Fokus utama tugas yang saya terima adalah terlibat dalam pengembangan dan pemeliharaan aplikasi berbasis web untuk sistem Sumber Daya Manusia (SDM) suatu perusahaan. Saya ditugaskan sebagai *full-stack developer* dan juga berkontribusi dalam pengembangan sebagian database yang digunakan dalam aplikasi tersebut. Tugas utama yang diberikan kepada saya adalah membangun modul *Applicant* pada aplikasi tersebut.

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Mendapat proyek terbaru yang akan diselesaikan secara kelompok, melakukan diskusi tentang rancangan awal pada platform web, dan menentukan pembagian tugas.
2	Fase inisial dalam pengembangan git atau ruang kerja. Membuat model contoh dari salah satu bagian proyek aplikasi berbasis web.
3	Merencanakan struktur database yang diperlukan untuk bagian proyek yang sedang dikerjakan, kemudian melakukan peninjauan terhadap tugas yang telah diselesaikan.
4	Pertemuan pada awal pekan ini digelar untuk membicarakan pembaruan pada situs web, selanjutnya melakukan konstruksi halaman sesuai dengan mock-up yang telah ditetapkan, dan menerapkan penggunaan database yang telah direncanakan.
5	Mengadakan pertemuan dengan klien untuk melakukan presentasi awal dan membicarakan kebutuhan yang belum terpenuhi.
6	Melakukan rapat untuk membicarakan pekerjaan ekstra yang akan diberikan kepada setiap anggota. Melanjutkan proyek dengan mengimplementasikan persyaratan yang diberikan oleh klien.
7	Menunggu tanggapan dari klien karena terdapat kekurangan informasi mengenai cara kerja situs web.
8	Melakukan proses <i>debugging</i> sekali lagi untuk memverifikasi perubahan yang baru dilakukan.
9	Mendapatkan umpan balik dari klien dan melanjutkan pekerjaan proyek berdasarkan masukan tersebut.
10	Menjalankan proses <i>debugging</i> pada output awal dari aplikasi web dan melakukan perbaikan cepat pada aplikasi tersebut.

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

11	Melakukan pertemuan kembali dengan klien untuk membahas fitur yang kurang dan melakukan demonstrasi. Menerapkan perubahan sesuai keinginan klien.
12	Mengimplementasikan tambahan perubahan yang diminta oleh klien, bersamaan dengan perubahan yang telah dilakukan pada minggu sebelumnya.
13	Mengadakan pertemuan untuk melakukan demonstrasi aplikasi dan melakukan pemasangan di server klien.
14	Menyelesaikan penambahan kecil yang diminta oleh klien tanpa mengubah struktur yang sudah ada pada situs web.
15	Menyelesaikan pekerjaan dan melakukan instalasi <i>website</i> agar dapat dipakai oleh <i>client</i> .
16	Melakukan pembaharuan source library yang digunakan pada <i>website</i> , dan menyelesaikan permintaan baru dari <i>client</i> setelah berjalannya aplikasi.

### 3.3 Uraian Pelaksanaan Magang

Pada bagian ini akan dijelaskan mengenai tugas-tugas apa saja yang telah ditugaskan dan dikerjakan sepanjang kerja magang.

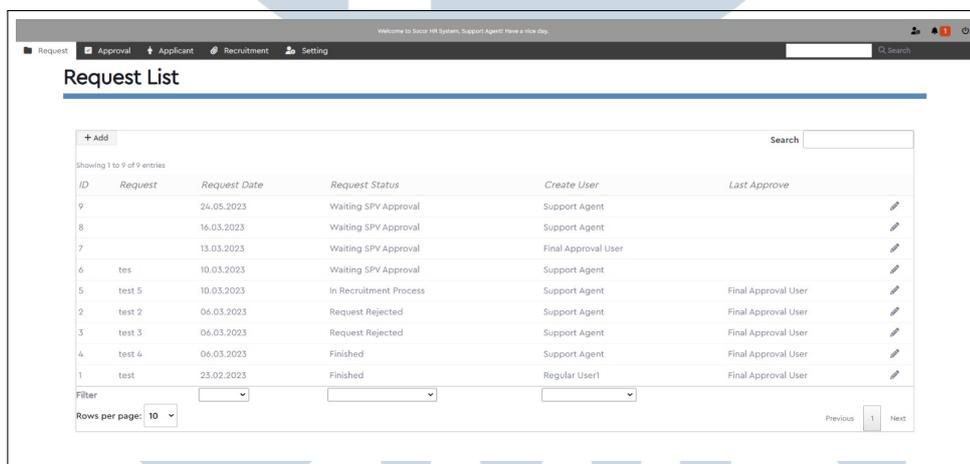
#### 3.3.1 Pembuatan tampilan dari mockup

Pembuatan *mockup* yang dilakukan adalah membuat tampilan *website* secara sederhana, sesuai dengan tampilan yang diberikan oleh atasan. Tampilan *mockup* tidak dikerjakan dari tahap awal dikarenakan atasan sudah mempersiapkan tampilan dari *website* yang akan dikembangkan, sehingga hanya perlu membuat *website* yang sudah menyerupai tampilan *mockup*. Tampilan pada *website* sedikit menyerupai dengan tampilan *website* yang sudah ada sebelumnya pada prosolution go! sehingga mempermudah dalam pengerjaan tampilan *website*. Tampilan yang sudah dibuat hanya bisa diklik tanpa melakukan suatu fungsi. Salah satu contoh tampilan yang dibuat berdasarkan *mockup* adalah modul request.



Gambar 3.1. mockup request Sucor HR

Gambar 3.1 tampilan *mockup* modul request yang sudah direvisi oleh atasan, dan yang nantinya akan digunakan untuk pembuatan tampilan situs. *mockup* halaman ini yang akan menjadi tampilan utama, pada saat pengguna berhasil masuk ke situs.

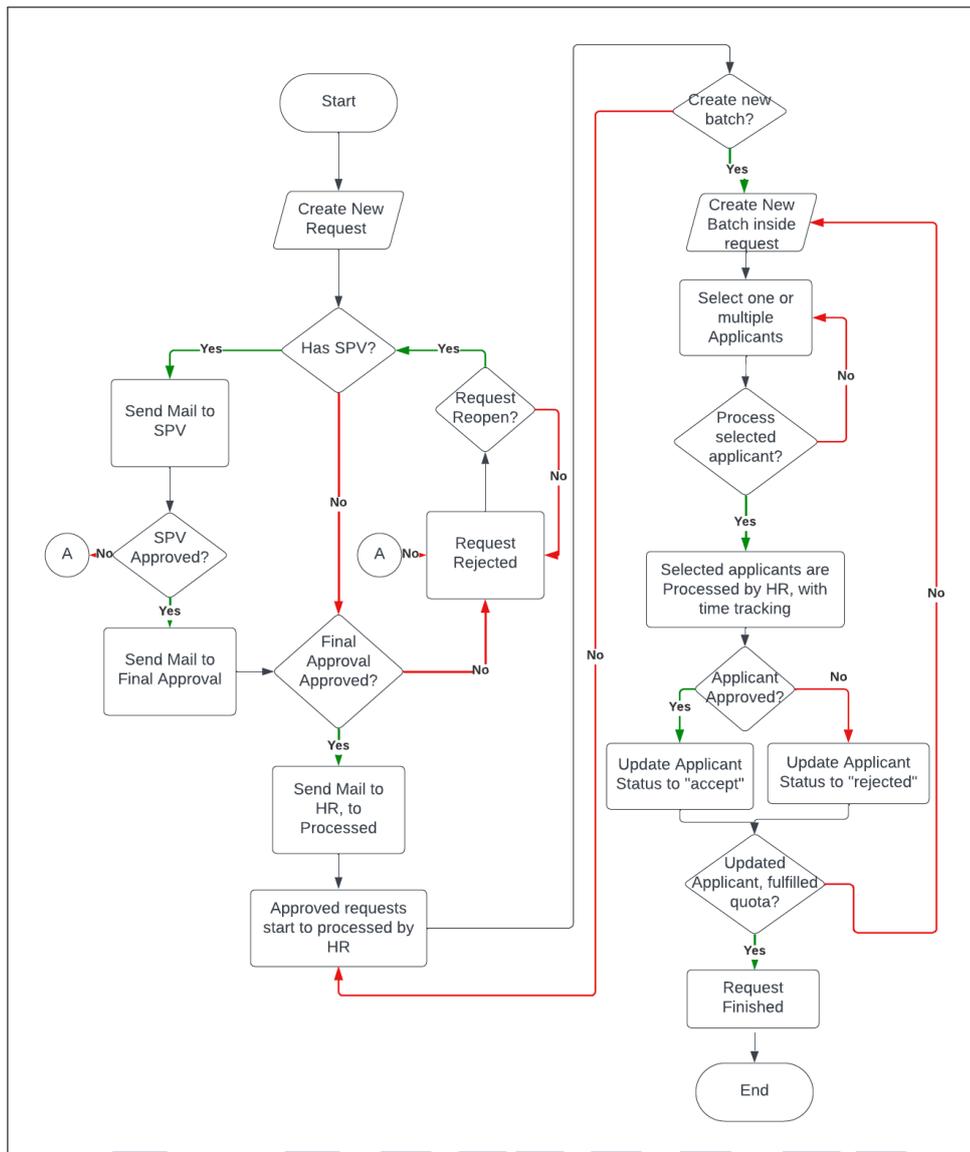


Gambar 3.2. hasil mockup request Sucor HR

Gambar 3.2 hasil tampilan akhir dari *mockup* modul request yang sudah berhasil dibangun, dan digunakan untuk pembuatan situs. Tampilan ini menggunakan plugin Datatables dari jQuery untuk menampilkan list permintaan.

### 3.3.2 Pembuatan flowchart website

*Flowchart* dibuat dengan menggunakan aplikasi open source yang tersedia yaitu lucidchart. *Flowchart* dibuat berdasarkan dengan alur dari *website* yang akan dikembangkan, *flowchart* berisi alur utama atau fungsi utama dari *website* ini.



Gambar 3.3. Flowchart Aplikasi Sucor HR

Gambar 3.3 menggambarkan *flowchart* dari situs yaitu dimulai pada saat *user* membuat permintaan baru untuk pembukaan lowongan pekerjaan atau magang, yang nantinya perlu dikonfirmasi dan diterima oleh atasan atau *supervisor*. Jika tahap tersebut telah selesai dilakukan maka akan dimulai tahap penerimaan karyawan baru yang dapat dilakukan dengan cara memulai *batch* baru dengan batasan 10 batch, jika lebih dari itu maka akan dibuka permintaan baru sesuai dengan yang pertama kali dibuat.

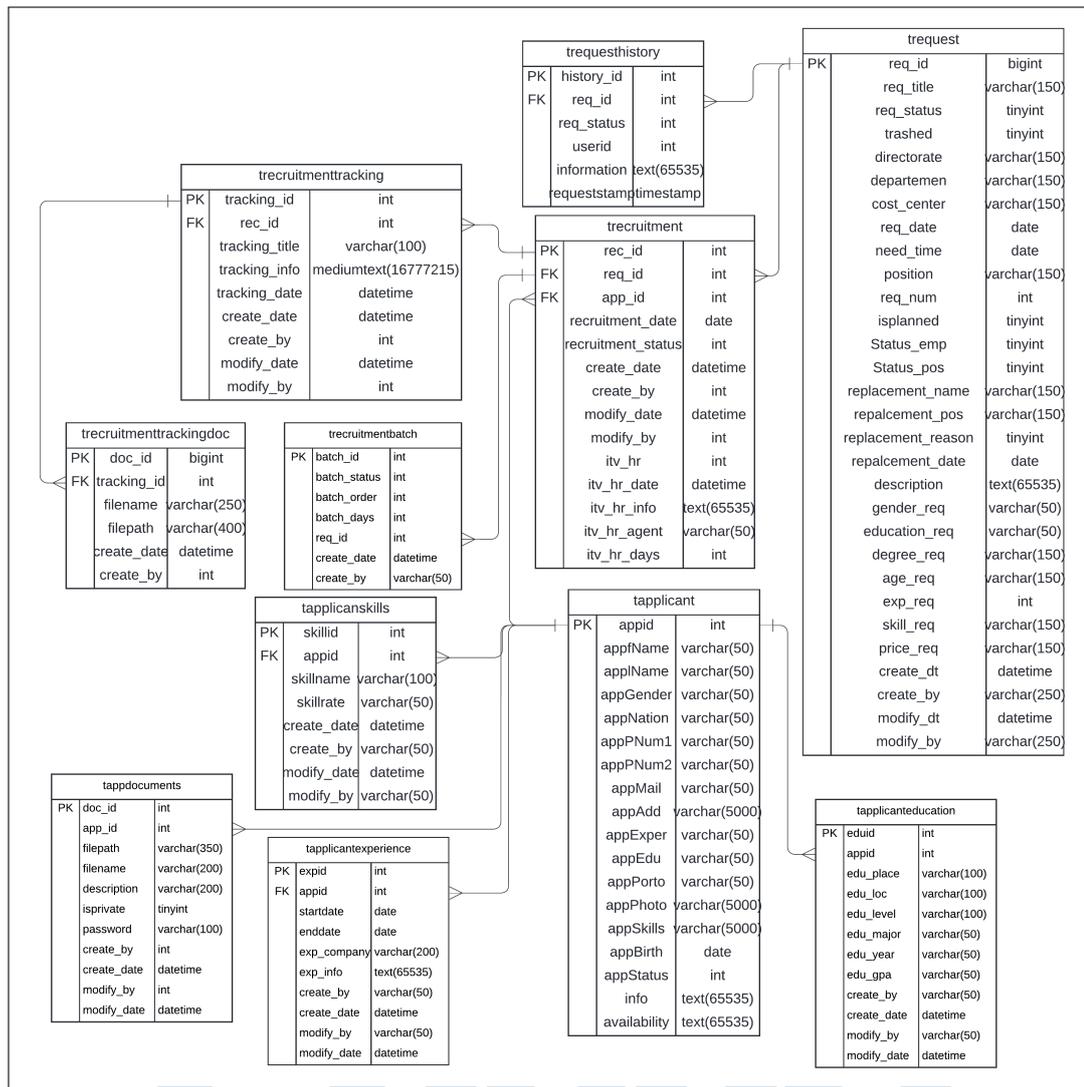
*Batch* hanya dapat diisi sesuai dengan ketentuan dari permintaan tersebut, berupa jumlah karyawan yang dicari, tanggal *deadline* dari permintaan tersebut, dan posisi dari karyawan yang dicari. Setelah calon karyawan telah dipilih

maka *user* yang membuat permintaan tersebut dapat menerima atau menolak calon karyawan yang telah dipilih, apabila diterima maka kuota atau jumlah dari karyawan diperlukan akan diperbarui, dan calon karyawan akan menerima notifikasi berupa email untuk *update* status dari penerimaan karyawan tersebut, seperti tahap wawancara, tahap negosiasi gaji, dan tahap penerimaan karyawan. Ketika kuota sudah terpenuhi maka permintaan akan ditutup dan dianggap sudah memenuhi permintaan.

### 3.3.3 Pembuatan database

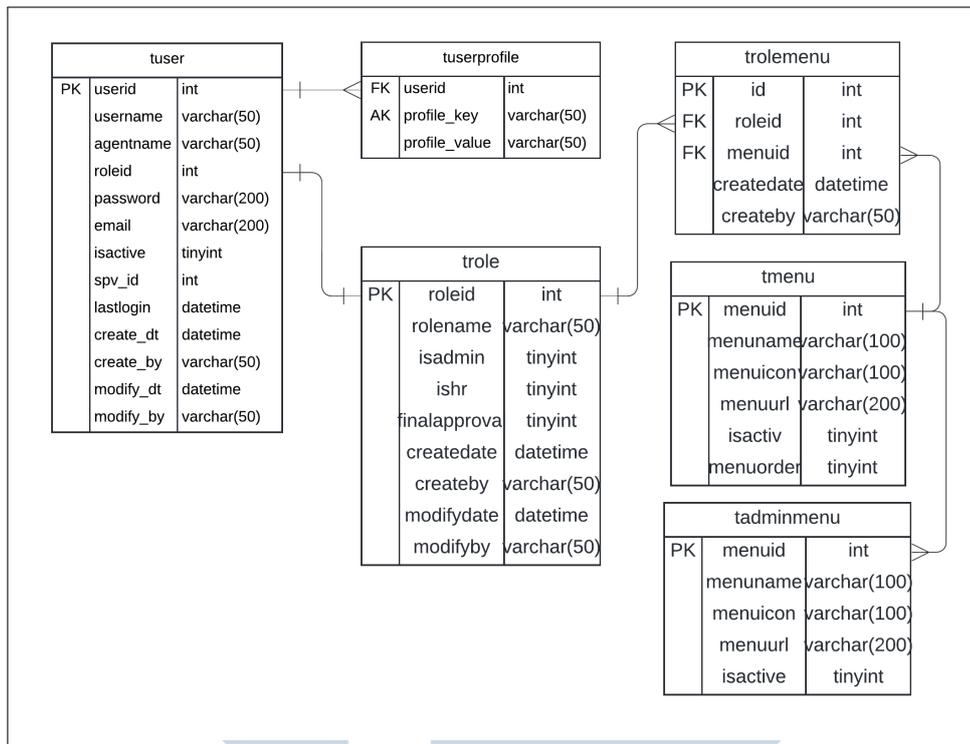
Pembuatan *database* dilakukan dengan cara diskusi dengan atasan dan para *developer* lainnya dengan melakukan perancangan secara lisan pada papan tulis. Nantinya, apabila adanya perubahan pada tabel *database* maka perubahan itu akan diperbarui dengan menggunakan script Alter. Pada pembuatan *database* digunakan SQL sebagai bahasa pemrograman untuk menyimpan dan memproses informasi.





Gambar 3.4. ERD recruitment dan applicant Sucor HR

Gambar 3.4 menjelaskan mengenai susunan *database* yang dipakai di dalam situs, khususnya pada modul utama seperti modul recruitment, modul applicant, dan modul request. Tabel *database* yang sudah dibuat pun telah diperiksa untuk memastikan bahwa ukuran, struktur, dan nilai yang dipakai sudah sesuai sehingga tidak memakan terlalu banyak *resources* pada server.



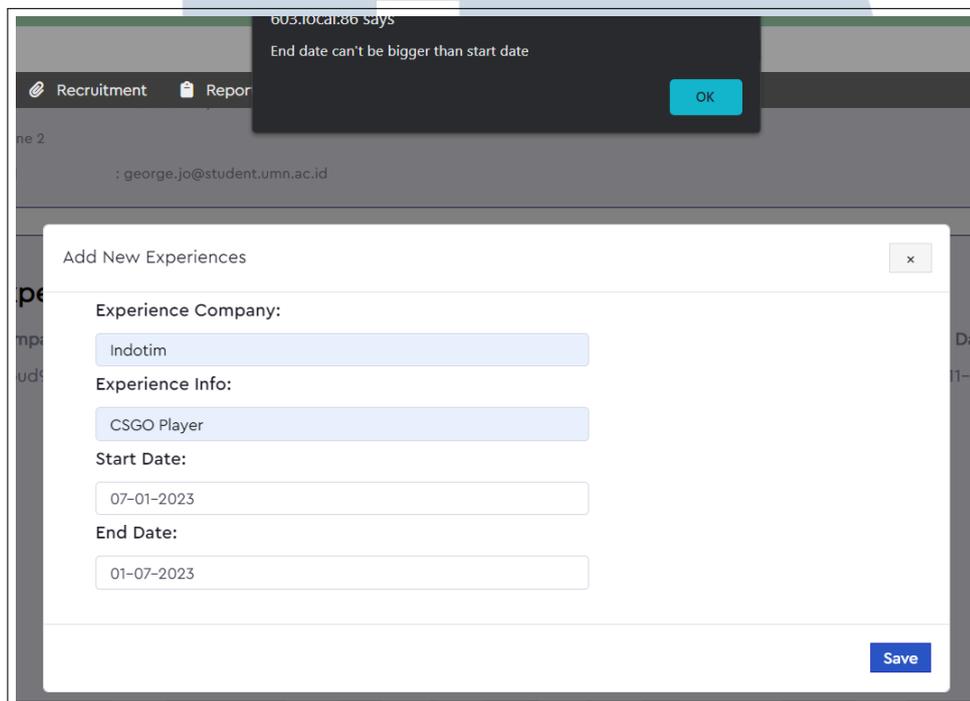
Gambar 3.5. ERD user dan menu Sucor HR

Gambar 3.5 menggambarkan mengenai ERD yang digunakan pada *user*, *user role* dan *menu website*. Seperti yang bisa dilihat *trole* memiliki dua hubungan dengan *trolemenu* yang berfungsi untuk mengatur menu yang akan ditampilkan ke pengguna, dan juga *tuser* yang berisikan daftar dari pengguna yang aktif maupun yang tidak aktif, dan *tuserprofile* berfungsi untuk menyimpan profil yang digunakan pengguna.

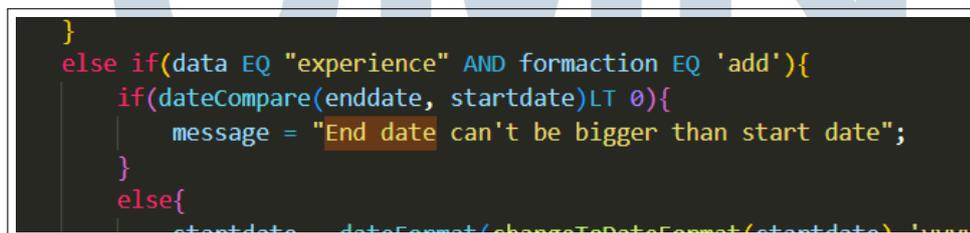
### 3.3.4 Melakukan debugging sederhana

Proses *debugging* dilakukan apabila terdapat *error* yang ditemukan pada saat pengembangan *website*, yang tidak diketahui asal awalnya. *Error* yang dimaksud berupa kesalahan yang dilakukan *developer* pada saat tahap pengembangan, seperti penggunaan fungsi klik pada tombol yang tidak bekerja apabila tombol tersebut ditekan berkali-kali maka proses *debugging* akan dilakukan untuk mencari letak kesalahan dan memperbaikinya. Proses *debugging* yang dilakukan dilakukan dengan memulai mencari letak sumber masalah, lalu setelah ditemukan pengembang harus mencari cara untuk mengatasi masalah tersebut, biasanya pada tahap ini akan meminta bimbingan *supervisor* untuk memastikan solusi yang akan diimplementasikan sudah sesuai dan efisien.

Salah satu contoh kasus yang dialami pada saat pengembangan situs adalah penggunaan format tanggal pada situs yang menyebabkan hasil *false positive* pada saat pengolahan data, kesalahan ini dapat terjadi apabila tanggal yang dimasukan memiliki hari yang sama dengan bulan, dengan tanggal mulai lebih besar dari tanggal selesai. Pada saat data tersebut disimpan maka muncul sebuah notifikasi bahwa data tersebut terbaca salah karena data tersebut dianggap memiliki tanggal mulai yang lebih besar dari tanggal selesai.



Gambar 3.6. Notifikasi masalah pada saat menyimpan



Gambar 3.7. Potongan source code yang menjadi penyebab

Seperti pada Gambar 3.6, terlihat bahwa tanggal yang dimasukan sudah benar yaitu 7 Januari 2023 hingga 1 Juni 2023, namun notifikasi tetap muncul. Setelah dilakukan *code tracing*, ditemukan bahwa kesalahan bersumber dari proses data, Gambar 3.7 Masalah ini akhirnya didiskusikan dengan atasan dikarenakan tidak ditemukan jawaban yang spesifik, atasan pun pernah mengalami hal yang

sama seperti ini dan merekomendasikan untuk mengubah teks yang didapat dari pemilihan tanggal sehingga diubah menjadi format tanggal. Metode yang dapat digunakan adalah dengan membaca teks secara manual dan memasukkan satu persatu sehingga menjadi format tanggal yang benar.

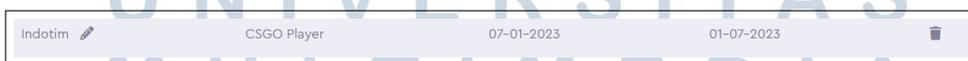
```
<cffunction name="changeToDateFormat" returntype="any" hint="Change date Format from dd.mm.yyyy to date variable">
  <cfargument name="sDate" required="Yes" type="string" hint="Date With Format dd.mm.yyyy or YYYYMMDD" />
  <!-- Define the local scope. -->
  <cfset var local = {} />
  <cfset local.arrDate = listToArray(arguments.sDate,"-") />
  <cfif arrayLen(local.arrDate) EQ 3>
    <cftry>
      <cfreturn createDate(local.arrDate[3], local.arrDate[2], local.arrDate[1]) />
      <cfcatch type="any">
        <cfreturn "" />
      </cfcatch>
    </cftry>
    <cfreturn createDate(local.arrDate[3], local.arrDate[2], local.arrDate[1]) />
  <cfelseif len(arguments.sDate) eq 8>
    <cftry>
      <cfreturn createDate(Left(arguments.sDate,4),mid(arguments.sDate,5,2),Right(arguments.sDate,2)) />
      <cfcatch>
        <cfreturn "" />
      </cfcatch>
    </cftry>
  <cfelse>
    <cfreturn "" />
  </cfif>
</cffunction>
```

Gambar 3.8. Solusi format tanggal

Gambar 3.8 menggambarkan serta menjelaskan solusi yang dibuat untuk mengatasi kesalahan dalam membaca tanggal. Dikarenakan teks yang didapat setelah memilih tanggal menghasilkan format MM-DD-YYYY sedangkan format yang diinginkan adalah DD-MM-YYYY, sehingga perlu dilakukan secara manual dalam memformat ulang tanggal.

```
else if(data EQ "experience" AND formaction EQ 'add'){
  if(dateCompare(changeToDateFormat(enddate), changeToDateFormat(startdate))LT 0){
    message = "End date can't be bigger than start date";
  }
  else{
```

Gambar 3.9. Solusi format tanggal diimplementasikan

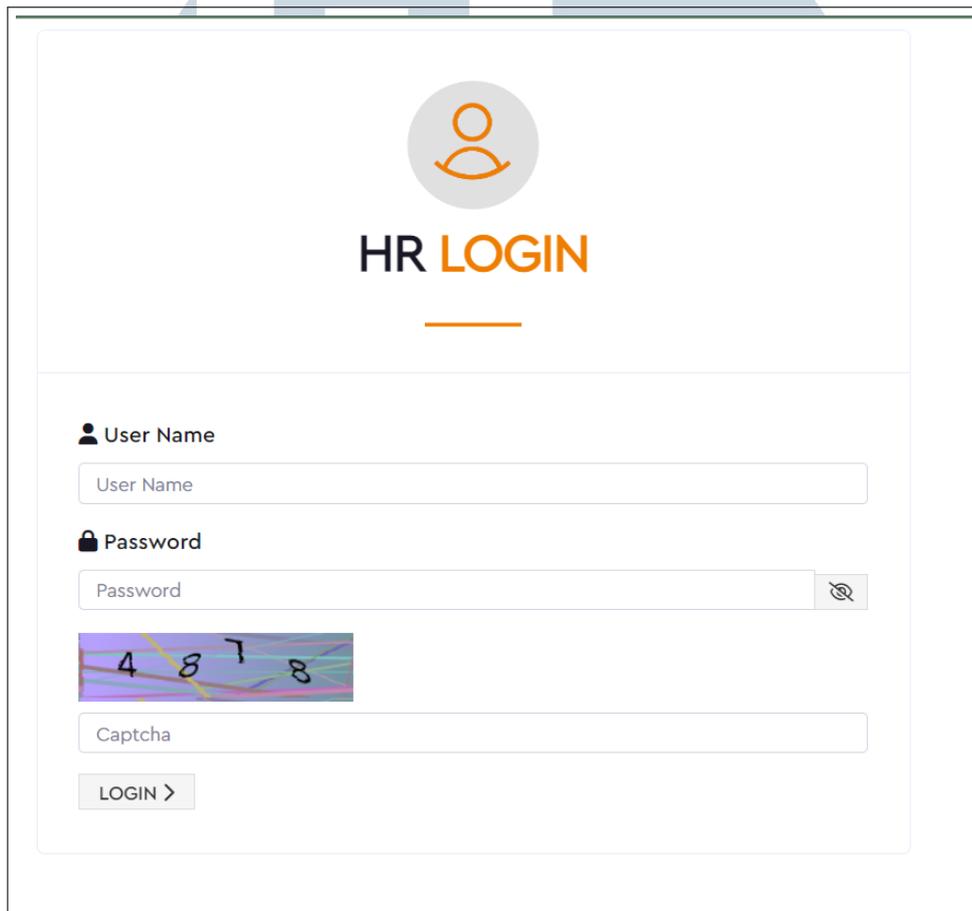


Gambar 3.10. Hasil dari implementasi solusi

Gambar 3.9 letak yang menjadi pengecekan tanggal juga diubah, untuk mengimplementasikan solusi yang sudah dibuat. Pada Gambar 3.10 solusi yang sudah diimplementasikan dapat memperbaiki notifikasi *false positive* yang sebelumnya dialami, dan data pun dapat disimpan.

### 3.3.5 Melakukan pembangunan situs web

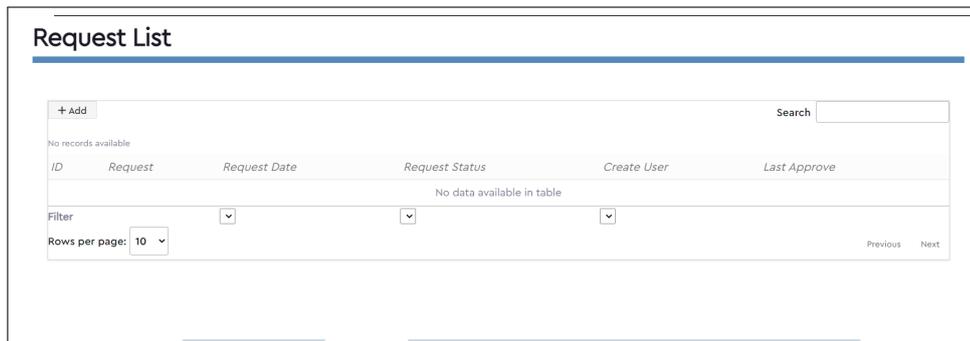
Pembangunan *website* dilakukan sesuai dengan modul yang dibagikan oleh *supervisor*. Pembangunan dilakukan pada modul "Request", "Approval", "Recruitment", dan "Applicant".



The image shows a web login interface for an HR application. At the top center, there is a circular icon containing a stylized person figure, with the text "HR LOGIN" below it. Underneath, there is a form with three input fields: "User Name", "Password", and "Captcha". The "Captcha" field contains a distorted image of the numbers "4 8 7 8". A "LOGIN >" button is positioned at the bottom of the form.

Gambar 3.11. Halaman login dari aplikasi sucor HR

Gambar 3.11 merupakan tampilan halaman pertama dari aplikasi ketika membuka *website* yaitu sebuah halaman login yang memerlukan *username* dan *password* dari pengguna yang sudah terdaftar, terdapat juga captcha yang perlu diisi terlebih dahulu sebelum masuk ke situs.

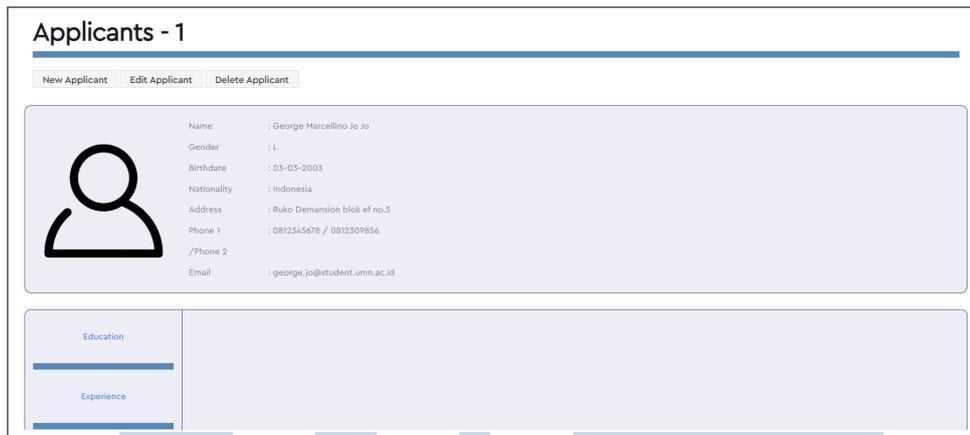


Gambar 3.12. Halaman utama dari aplikasi sucor HR

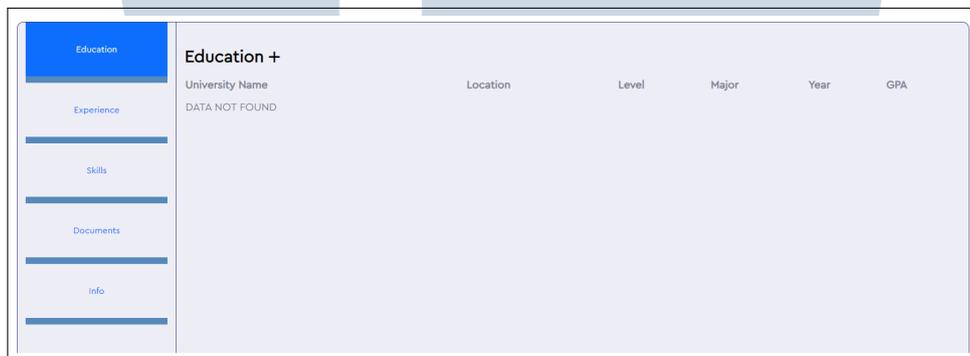
Gambar 3.12 menjelaskan halaman pertama dari aplikasi ketika melewati halaman login yaitu halaman *request*, pada modul ini digunakan untuk membuat permintaan baru dari pengguna biasa dan menampilkan nya dalam bentuk list. Nantinya modul ini juga dapat dilihat oleh *supervisor* untuk membuat permintaan yang nantinya akan langsung ke tahap *Recruitment* tanpa melewati tahap penerimaan *Request*.

Gambar 3.13. Bentuk form dari halaman request aplikasi sucor HR

Gambar 3.13 menggambarkan bentuk form dari modul *request* yang berisi mengenai data-data yang diperlukan untuk pembukaan lowongan pekerjaan baru seperti nama dari permintaan tersebut, tanggal diperlukannya karyawan baru, posisi apa yang diperlukan, skill apa yang diperlukan untuk memenuhi posisi tersebut, departemen/divisi apa yang meminta, jumlah karyawan yang diperlukan, jenis kelamin dari karyawan, minimum pengalaman yang dimiliki, dan pendidikan terakhir. Data-data yang diperlukan tersebut sudah sesuai dengan apa yang dimintakan oleh *client* yang bersifat kustomisasi situs.



Gambar 3.14. Halaman applicant aplikasi sucor HR



Gambar 3.15. Halaman applicant details edukasi aplikasi Sucor HR



Gambar 3.16. Halaman applicant details pengalaman aplikasi Sucor HR

Gambar 3.14 menggambarkan mengenai tampilan halaman modul *applicant*. Pada modul ini calon karyawan baru yang dimasukan oleh HR akan digunakan untuk proses rekrut, yang dapat digunakan setelah permintaan karyawan baru diterima oleh atasan di dalam perusahaan. Nantinya calon karyawan yang sudah terdata di situs, akan disimpan data-data berupa edukasi pada Gambar 3.15, pengalaman pada Gambar 3.16, skill, dokumen seperti CV atau penghargaan, dan

info tambahan seperti gaji yang diminati atau ketersediaan bekerja secara WFH (*Work From Home*) atau WFO (*Work From Office*).

The screenshot shows a web form titled "Add New Applicant". The form contains the following fields and options:

- First Name:** A text input field containing "George Marcellino Jo".
- Last Name:** An empty text input field.
- Gender:** Radio buttons for "L" and "P".
- Birthdate:** An empty text input field.
- Nationality:** An empty text input field.
- Phone Number:** An empty text input field.
- Phone Number 2:** An empty text input field.
- Email:** An empty text input field.

Gambar 3.17. Bentuk form dari halaman applicant aplikasi sucor HR

Gambar 3.17 menggambarkan bentuk form pada modul *applicant* yang digunakan untuk memasukkan atau mengubah data calon karyawan yang hanya dapat diakses oleh pengguna HR.

The screenshot shows a web page titled "Approval List". It features a search bar at the top right. Below the search bar, there is a message "No records available". A table with the following columns is shown:

ID	Request	Request Date	Request Status	Create User	Last Approve
No data available in table					

Below the table, there are filter and pagination controls. The "Rows per page" is set to 10. There are "Previous" and "Next" buttons for navigation.

Gambar 3.18. Tampilan halaman approval aplikasi sucor HR

Gambar 3.18 menggambarkan bentuk halaman dari modul *approval* yang memiliki kemiripan dengan modul *request*. Pada modul ini pengguna yang memiliki peran sebagai atasan atau sebagai direktur dapat mengakses dan menerima

permintaan-permintaan baru dari tiap divisi/departemen. Setelah tahan ini selesai dilakukan maka akan berlanjut ke tahap pengrekrutan karyawan baru.

ID	Request	Request Date	Applicant	Last Approve	Request Status
25	Test request	27/11/2023	0/1	Jessica Wijaya	In Progress

Gambar 3.19. Tampilan halaman recruitment aplikasi sucor HR

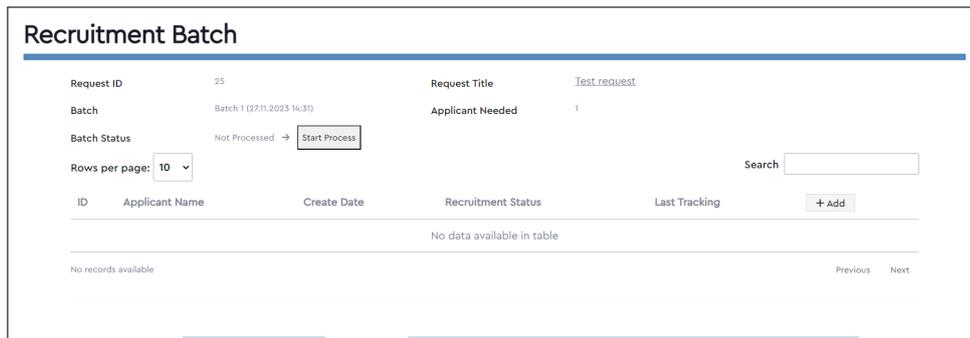
Gambar 3.19 menggambarkan bentuk halaman dari modul *recruitment* yang memiliki kemiripan dengan tampilan modul *request* dan *approval*, dapat diakses dengan klik salah satu *recruitment list*. Pada modul ini pengguna yang memiliki peran sebagai HR akan memiliki wewenang untuk memasukkan calon-calon karyawan kedalam *recruitment* dengan menggunakan sistem *batch*, sehingga di dalam suatu permintaan rekrut bisa memiliki maksimal 10 *batch*. Apabila kuota sudah terpenuhi maka permintaan akan ditutup dan dianggap sebagai *done*, sementara jika belum, pengguna dapat memilih untuk membuka kembali atau hanya membiarkan kuota tersebut tidak terisi namun dengan status *done*.

Request ID	25	Request Title	Test request
Final Approval Date	27/11/2023 14:29 by Jessica Wijaya	Request Quota	0/1

Recruitment Batch +	Number Applicant	Number Approved	Status

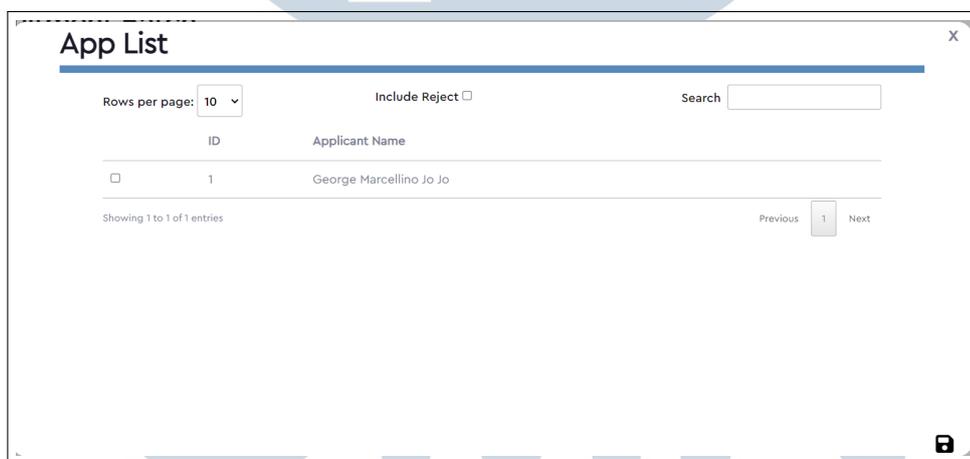
Gambar 3.20. Tampilan halaman recruitment detail aplikasi Sucor HR

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.21. Tampilan halaman recruitment batch aplikasi Sucor HR

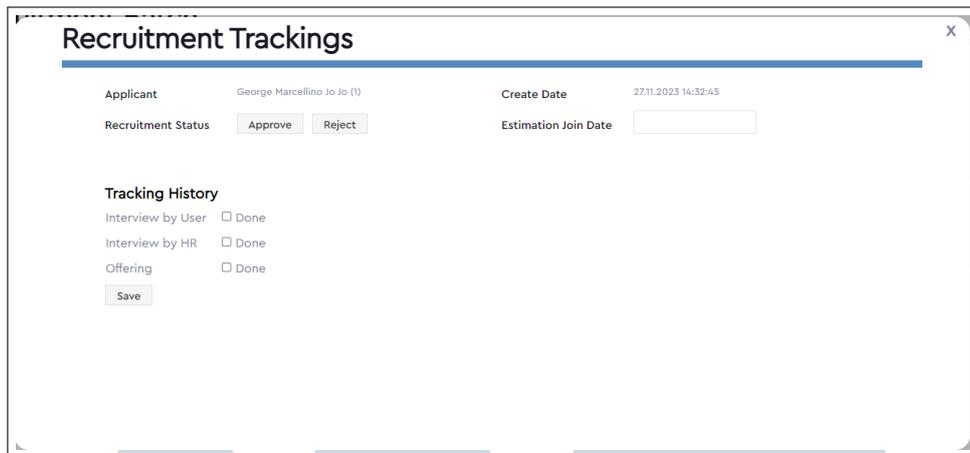
Gambar 3.20 menggambarkan bentuk halaman detail dari modul *recruitment*, yang berisi informasi dari kuota yang diperlukan, tanggal pembukaan lowongan, dan juga data dari permintaan yang diperlukan dalam perekrutan. Dibagian bawah halaman terdapat *batch* yang nantinya dapat diklik untuk menambahkan *batch*. Apabila diklik *batch* yang sudah dibuat, maka akan muncul detail dari *batch* tersebut, berupa karyawan yang diperlukan, status, tanggal *batch*, dan nama-nama pelamar. Seperti pada Gambar 3.21.



Gambar 3.22. Tampilan halaman recruitment aplikasi Sucor HR



Gambar 3.23. Tampilan halaman recruitment batch aplikasi Sucor HR



Gambar 3.24. Tampilan halaman recruitment trackings aplikasi Sucor HR

Apabila tombol "add" diklik pada halaman *recruitment batch* Gambar 3.21, maka akan muncul list dari nama-nama pelamar yang sudah tersimpan, seperti pada Gambar 3.22. Apabila ada applicant yang terpilih, dan disimpan maka akan muncul pada halaman *recruitment batch* Gambar 3.23. Setelah itu apabila *batch* diproses, maka pelamar dapat melalui proses pelamaran kerja, yang status dari proses tersebut dapat dilihat pada recruitment trackings pada Gambar 3.24.

### 3.3.6 Melakukan pembaharuan library

Pembaharuan *library* dilakukan setelah 4 bulan yang terhitung dari awal pengembangan *website*, apabila pada saat pengecekan *library* terdapat versi terbaru maka akan dilakukan *update* agar tidak adanya ketertinggalan versi *library*. Beberapa *library* yang sudah diperbaharui yaitu FontAwesome versi 6.4.2, jQuery versi 3.7.1, Bootstrap versi 5.3.2 dan DataTables versi 1.13.7, yang sudah dilakukan pada saat pembangunan situs yaitu tanggal 15 Oktober 2023.

## 3.4 Kendala dan Solusi yang Ditemukan

Pada bagian ini akan dijelaskan mengenai kendala yang dialami selama bekerja magang serta solusi yang didapat setelahnya.

### A. Kendala yang ditemukan

1. Pembelajaran bahasa pemrograman.

Indotim menggunakan bahasa pemrograman Coldfusion, yang merupakan

bahasa yang kurang populer dan sulit untuk menemukan dokumentasi terkini. Oleh karena itu, mencari sumber pembelajaran yang terpercaya dari internet menjadi tugas yang tidak mudah.

2. Komunikasi dengan senior.

Di perusahaan Indotim, komunikasi jarang dilakukan secara lisan, melainkan melalui aplikasi yang hanya dapat diakses oleh karyawan tetap. Kondisi ini menyebabkan interaksi antar karyawan secara lisan di kantor Indotim menjadi kurang umum. Oleh karena itu, bagi karyawan magang yang tidak memiliki akses ke chatroom, disarankan untuk lebih proaktif dan mengajukan pertanyaan secara langsung kepada senior atau *supervisor* yang bersangkutan.

3. Performa aplikasi

Saat modul "Applicant" dibuat, terjadi penurunan kinerja saat menampilkan data karena kesalahan dalam menerapkan algoritma yang dapat menampilkan data berdasarkan kolom yang diklik.

4. Penampungan requirements

Karena klien memiliki pengetahuan terbatas mengenai sistem HR/CRM, terdapat tantangan dalam mengembangkan aplikasi sesuai dengan harapan klien. Hal ini disebabkan oleh pemahaman yang terbatas dari klien terhadap gambaran keseluruhan dari aplikasi yang diinginkan.

## **B. Solusi yang diberikan**

1. Pembelajaran bahasa pemrograman.

Untuk menguasai bahasa pemrograman Coldfusion, diperlukan pencarian sumber yang kredibel dan menyediakan dokumentasi lengkap tentang Coldfusion. Selain mengandalkan dokumentasi, penting juga untuk mengasah keahlian melalui praktik penggunaan bahasa pemrograman ini, termasuk pembuatan demo dengan menggunakan Coldfusion. Melalui pembelajaran proyek yang sudah diselesaikan atau yang sedang dalam pengembangan, dapat meningkatkan pemahaman secara langsung mengenai cara kerja bahasa pemrograman tersebut. Tentu saja, rasa ingin tahu individu juga menjadi faktor penting dalam memahami bahasa pemrograman ini.

2. Komunikasi dengan senior.

Sebagai karyawan magang, dianjurkan untuk lebih proaktif dan menunjukkan

minat yang kuat terhadap proses kerja atau pembelajaran dalam suatu proyek. Alternatifnya, karyawan dapat meminta informasi kontak pribadi dari atasan atau supervisor, sehingga jika orang tersebut tidak berada di lokasi fisik, komunikasi masih dapat dilakukan secara daring melalui aplikasi obrolan.

### 3. Performa aplikasi

Proses peninjauan kode dilakukan bersama atasan, termasuk pembuatan atau penerapan logika baru menggunakan algoritma yang disarankan oleh atasan. Jika terdapat algoritma yang diketahui dan dianggap dapat dianjurkan, hal tersebut dapat dibahas dan direkomendasikan kepada atasan.

### 4. Penampungan requirements

Pengetahuan yang dimiliki oleh pengembang atau pemimpin TI terkait aplikasi CRM/HR menjadi krusial, sehingga persyaratan tidak hanya bersumber dari klien tetapi juga berasal dari pengalaman internal. Selain memiliki pengetahuan, pertemuan dengan klien untuk mendapatkan persyaratan melalui diskusi juga sangat diperlukan. Karena persyaratan memiliki peran penting, penanganan kendala terkait hal ini menjadi tanggung jawab para atasan di dalam perusahaan.

