

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Praktik kerja magang dilaksanakan di PT ABC yang dimana strukturnya mengikuti PT XYZ dan penempatannya disesuaikan dengan kebutuhan perusahaan yaitu pada divisi *IT & Maintenance* sebagai *Junior Java Programmer*. Kerja Magang dibimbing oleh karyawan PT ABC & Manager di divisi *IT & Maintenance* PT ABC.

Proses rancang bangun aplikasi seperti *flow* sistem dan tampilan sistem diinstruksi oleh pembimbing lapangan di PT ABC secara langsung atau tatap muka sampai berjalan sesuai dengan kebutuhan departemen. Pembuatan sistem dilakukan secara mandiri sesuai instruksi pertama dan koordinasi dengan pembimbing lapangan secara langsung atau tatap muka pada bagian-bagian yang tidak dimengerti terutama di bagian *database* dan konfirmasi *testing program*. Koordinasi mengenai hal-hal yang tidak berhubungan dengan pembuatan aplikasi data karyawan dan sistem *pivot table* dilakukan dengan menggunakan WhatsApp dengan HRD PT XYZ.

3.2 Tugas yang Dilakukan

Selama 40 hari kerja, terdapat beberapa tugas yang telah dilaksanakan yaitu proses belajar java selama 1 minggu, belajar membuat berbagai aplikasi sederhana selama 2 minggu. Membuat sistem *pivot table* selama 1 bulan, dan membuat presentasi dan melakukan presentasi sistem *pivot table* selama 1 minggu yang dapat dilihat pada Tabel 3.1.

Tabel 3.1 Jadwal Kerja Magang

Nama Kegiatan	Minggu ke-							
	1	2	3	4	5	6	7	8
Pengenalan Lingkungan Kerja								
Persiapan Peralatan Kerja								
Pembelajaran Sistem Kerja								
Pembuatan Aplikasi Data Karyawan								
Pembuatan Sistem <i>Pivot Table</i>								
<i>Bug Fixing</i> Sistem <i>Pivot Table</i>								
Revisi Sistem								
Dokumentasi								
Presentasi								

Perkenalan lingkungan kerja dilakukan pada minggu pertama. Pada hari pertama, dilakukan pengenalan kepada seluruh staf dan karyawan di PT ABC yang didampingi oleh *Human Resource* PT ABC. Selanjutnya, diberikan buku petunjuk mengenai tata laksana bekerja pada PT ABC dan buku petunjuk metode pelaksanaan kerja yang berjalan pada PT ABC. Pada hari kedua, dilakukan pembiasaan terhadap sistem yang berjalan di PT ABC.

Persiapan peralatan kerja dilakukan pada minggu pertama. Pada hari pertama, laptop diberikan untuk melakukan setup aplikasi dan *email* kantor. Aplikasi yang disiapkan berupa aplikasi *chatting local* PT ABC dan *email local* PT ABC. Selain itu, perlu juga membuat *user* untuk windows dan mempersiapkan jaringan LAN dan memastikan bahwa dapat terhubung ke jaringan *server local* PT ABC, *database dummy* PT XYZ, dan *internet*. Apabila windows yang digunakan perlu untuk di-*upgrade* atau dibutuhkan untuk meng-*install* aplikasi yang umum digunakan, *contact IT Support* untuk bantu *install*. Pada hari kedua dilakukan *install* aplikasi yang akan digunakan dalam pembuatan *project* yakni *postgre SQL*, *JBoss*, dan *Eclipse*.

Pembelajaran sistem kerja dilakukan pada minggu pertama. Pada hari kedua, dilakukan pembelajaran *java basic* yang mencakup *modifier types, operator, loops, decision making, number type, character, string, arrays, date & time, dan exception* serta PostgreSQL, JSF, Spring, dan XHTML. Pada hari ketiga, dilakukan pembelajaran *object oriented program* yang mencakup *inheritance, polymorphism, overriding, overloading, encapsulation, packages, interface, dan abstract* serta *Basic PrimeFaces*. Pada hari keempat, dilakukan pembelajaran *user interface* yang mencakup *swing* serta *PrimeFaces with Bean*. Pada hari kelima, dilakukan pembelajaran J2EE(*web-based application*) yang mencakup *basic servlet, server side, java servlet, servlet configuration* serta *PrimeFaces to Postgre*.

Pembelajaran pembuatan sistem dilakukan di minggu kedua. Pada kegiatan ini, dilakukan pembuatan contoh aplikasi yang dibuat berdasarkan apa yang dipelajari di minggu pertama. Aplikasi yang dibuat adalah aplikasi data karyawan yang menampilkan data-data karyawan beserta deskripsinya. Data yang diambil untuk menampilkan data karyawan diambil dari *database dummy* PT XYZ.

Pembuatan sistem dan *bug fixing* dilakukan pada minggu ketiga sampai minggu keenam. Pada hari pertama dilakukan diskusi dengan manager PT ABC yang memberi *project* magang. Diskusi mencakup bentuk dari sistem, *flow* dari sistem, dan *requirement* yang diinginkan pada sistem. Pada minggu ketiga hari kedua hingga hari kelima, dilakukan pembuatan *layout* sistem dan *flowchart* sistem. Pada minggu keempat hingga keenam, dilakukan pembuatan sistem dan *bug fixing* secara langsung ketika *bug* muncul.

Revisi sistem dilakukan dilakukan pada minggu ketujuh. Revisi sistem mencakup perubahan-perubahan fungsi yang ingin ditambahkan pada sistem *pivot table* seperti *Statistic Function*, *Renderer*, dan penambahan fungsi seperti *export* ke dalam bentuk CSV atau XLSX. Selain fungsi, *layout* dari sistem juga diubah sehingga *user* dapat menggunakan sistem dengan nyaman.

Dokumentasi sistem dilakukan pada minggu kedelapan. Pada kegiatan kali ini dilakukan dokumentasi cara penggunaan sistem dan pembuatan sistem secara singkat serta laporan waktu *query* yang dibutuhkan untuk menyelesaikan satu *task*. Dokumentasi terdiri dari dua *file* yakni ppt dan xlsx. File ppt mencakup *screenshot* sistem yang telah dibuat dan penjelasan singkat terkait penggunaan sistem beserta *screenshot coding* beserta penjelasan *step-by-step* dari *coding* tersebut.

3.3 Uraian Pelaksanaan Kerja Magang

Model pengembangan sistem *Software Development Life Cycle* (SDLC) diterapkan sebagai bentuk yang dilaksanakan selama praktik kerja magang. Metode SDLC digunakan karena pencerminan kepraktisan rekayasa, yang bisa membuat kualitas software tetap terjaga dan proses pemeliharannya lebih mudah (Billah, 2019).

3.3.1. Proses Pelaksanaan

Proses pembuatan sistem dibuat dengan user requirements, perancangan aplikasi, dan implementasi.

A. User Requirements

User requirements merupakan seluruh kebutuhan aplikasi yang diminta oleh perusahaan mengenai aplikasi yang dibuat selama magang berlangsung.

A.1. Kebutuhan Aplikasi Data Karyawan

Perusahaan meminta agar aplikasi data karyawan yang dibangun dapat melakukan beberapa fitur sebagai berikut.

1. *User* dapat menambah dan melakukan *input* data karyawan berupa nama, alamat, jenis kelamin, dan lulusan.
2. *User* dapat melakukan *reset* seluruh *field input* data.
3. *User* dapat melakukan *edit* dan *delete* data.
4. Pembuatan sistem menggunakan gabungan dari *procedural* dan OOP.

A.2. Kebutuhan Sistem Pivot Table

Perusahaan meminta agar sistem *pivot table* yang dibangun dapat melakukan beberapa fitur sebagai berikut.

1. *User* dapat langsung masuk ke dalam sistem tanpa harus melakukan *login*.
2. *User* dapat memilih cabang yang diinginkan.
3. *User* dapat memilih cakupan *logic* yang diinginkan.
4. *User* dapat membatasi banyaknya *data* yang muncul sesuai panjang hari.
5. *User* tidak perlu memasukkan kriteria pencarian(sudah ada *default*).
6. *User* dapat memilih hari mulai *data* ditampilkan.
7. *User* dapat menampilkan *pivot table* yang *interactive*.

8. *User* dapat mengubah *renderer* dan memiliki banyak *renderer*.
9. *User* dapat mengubah *statistic function* dan memiliki banyak *statistic function* dan dapat di *sorting*.
10. *User* dapat menarik *attributes* ke kolom atau baris.
11. *User* dapat melakukan *download table* dalam bentuk csv atau xlsx dan tanpa download ke dalam bentuk csv.
12. *User* dapat melakukan *convert table* tanpa *download* ke dalam bentuk csv.
13. Pembuatan sistem menggunakan gabungan dari OOP dan *Procedural*.

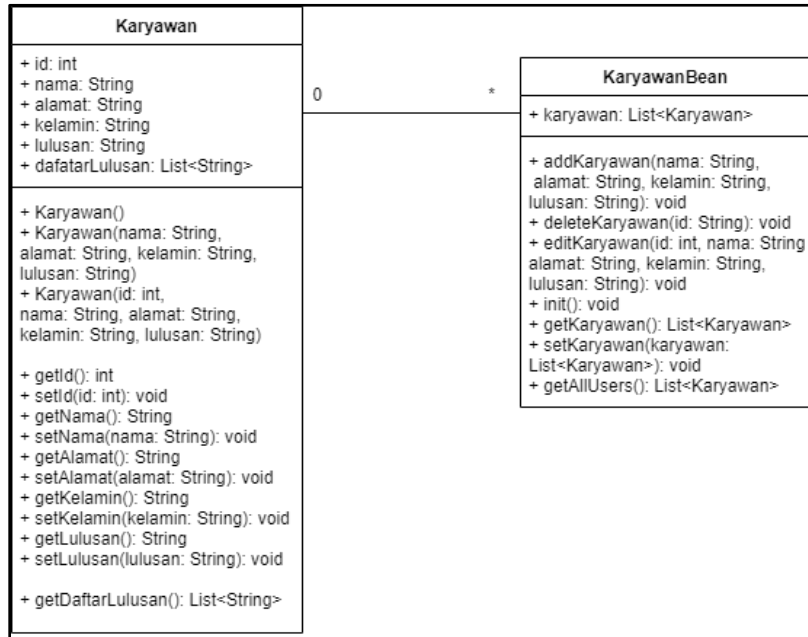
B. Perancangan Aplikasi

Perancangan aplikasi dibuat dengan menggunakan *uml* dan rancangan antarmuka.

B.1. Class Diagram

Class diagram merupakan perancangan class pada sistem data karyawan dan *pivot table* selama magang berlangsung.

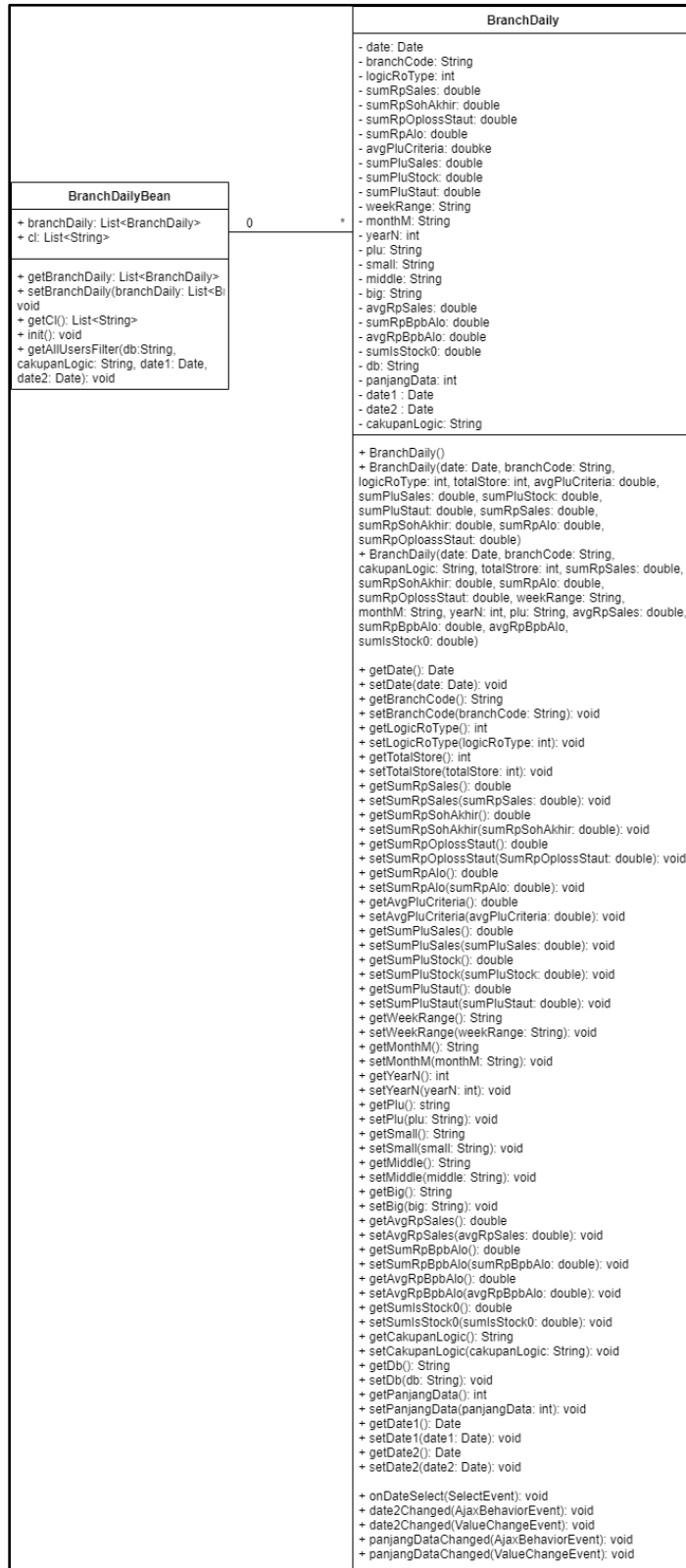
B.1.1. Class Diagram Data Karyawan



Gambar 3.1 *class diagram* data karyawan

Gambar 3.1 merupakan *class diagram* dari data karyawan yang terdiri dari *class* Karyawan dan KaryawanBean. *Class* Karyawan merupakan objek yang digunakan oleh KaryawanBean sebagai *list*. KaryawanBean memiliki fungsi-fungsi yang digunakan pada aplikasi data karyawan seperti tambah karyawan, edit karyawan, hapus karyawan, hingga menampilkan seluruh data karyawan.

B.1.2. Class Diagram Pivot Table



Gambar 3.2 class diagram Pivot Table

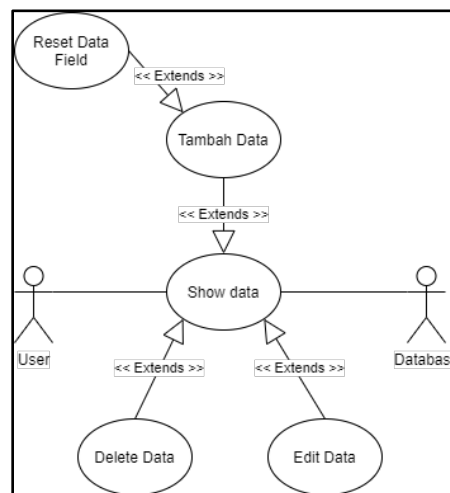
Gambar 3.2 menggambarkan *class diagram* dari sistem *pivot table* yang terdiri dari *BranchDaily* dan *BranchDailyBean*. *BranchDaily* merupakan objek yang digunakan pada sistem *pivot table* sebagai penampung kriteria pencarian, penampung *value* dari *attributes*, dan penampung *default* cabang pada kriteria pencarian.

BranchDaily memiliki 5 fungsi yang digunakan pada kriteria pencarian yang terbagi menjadi 5 *function* yakni, *onDateSelect(SelectEvent)*, *date2Changed (AjaxBehaviorEvent)*, *date2Changed (ValueChangeEvent)*, *panjangDataChanged (AjaxBehaviorEvent)*, dan *panjangDataChanged (ValueChangeEvent)*. Kelima fungsi ini merupakan fungsi yang akan mengubah *date1* dengan mengurangi *date2* dengan *date1*.

B.2. Use Case Diagram

Use case diagram yang dibuat selama magang adalah perancangan fungsi-fungsi yang akan dibuat pada data karyawan dan *pivot table*.

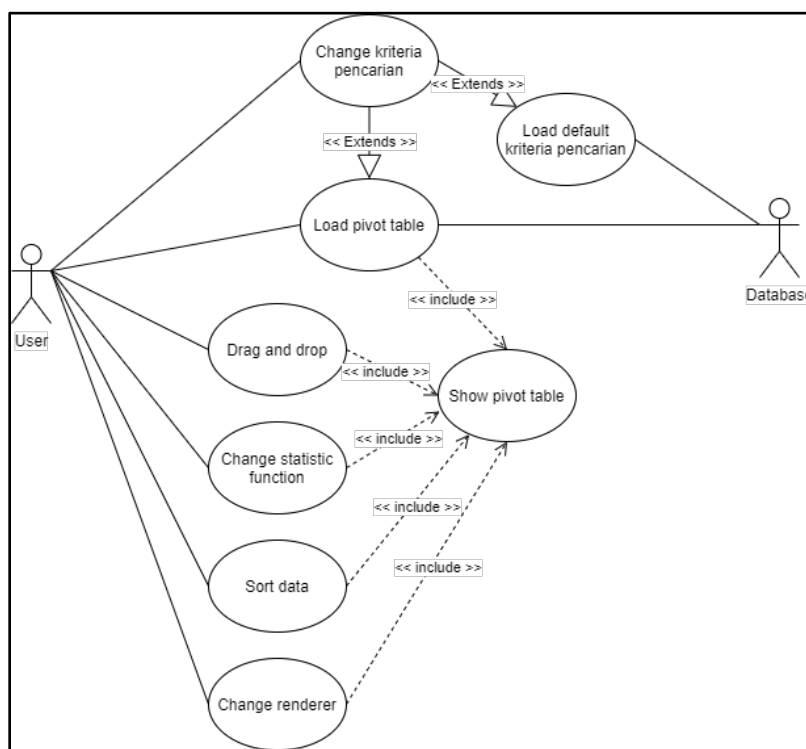
B.2.1. Use Case Diagram Data Karyawan



Gambar 3.3 *use case diagram* data karyawan

Gambar 3.3 menggambarkan *use case diagram* data karyawan yang memiliki 2 *actor* yaitu *user* dan *database*. *User* dapat menampilkan data dan terdapat 3 fungsi primer dan 1 fungsi sekunder. Fungsi primer merupakan *user* dapat melakukan penambahan data, *delete* data, dan *edit* data. Fungsi sekunder merupakan *user* dapat melakukan *reset* seluruh *field input* ketika *user* menambah data.

B.2.2. Use Case Diagram Pivot Table



Gambar 3.4 *use case diagram pivot table*

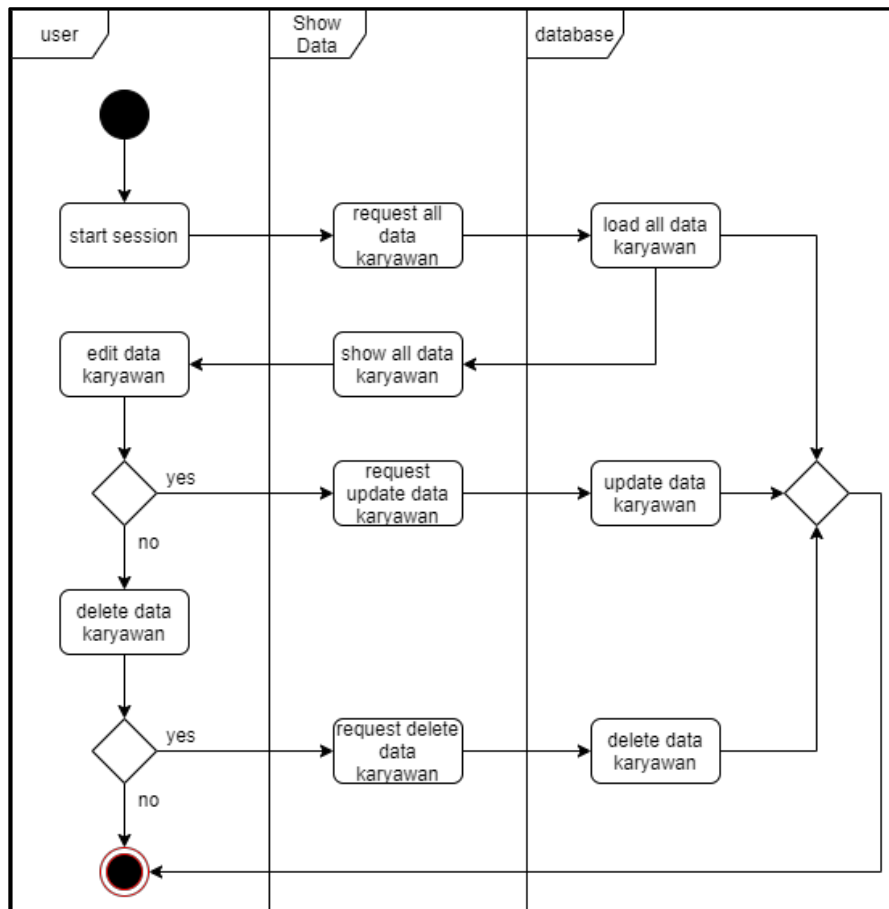
Gambar 3.4 menggambarkan *use case diagram pivot table* yang memiliki 2 *actor* yaitu *user* dan *database*. *User* dapat mengubah kriteria pencarian, menampilkan *pivot table*, memindahkan *unused attributes* dengan *drag and drop*, mengubah *statistic function*, *sorting* tabel pada *pivot table*, dan mengubah *renderer*. *Database* dapat melakukan *load default* kriteria pencarian, dan *load attribute pivot table*.

B.3. Activity Diagram

Activity Diagram adalah *flow* dari sistem data karyawan dan pivot table dari awal hingga aplikasi berhenti yang dibagi berdasarkan fungsinya.

B.3.1. Activity Diagram Data Karyawan

1. Data Karyawan – *Show Data*



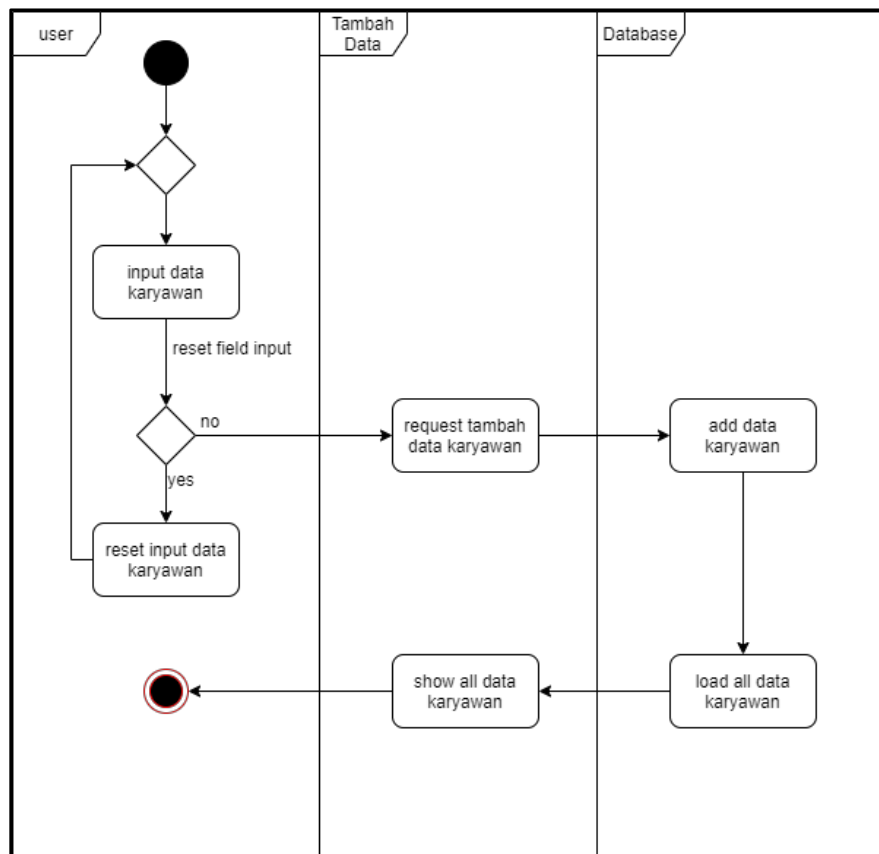
Gambar 3.5 *activity diagram* data karyawan – *show data*

Gambar 3.5 merupakan *activity diagram* data karyawan yang berfungsi untuk menampilkan seluruh data karyawan, melakukan *edit* data karyawan, dan melakukan *delete* data karyawan. Pertama kali user membuka aplikasi, sistem akan melakukan *request* ke *database* untuk melakukan *load* seluruh data karyawan dan

menampilkannya ke *user*.

Edit data karyawan dilakukan dengan cara melakukan *input* pada data karyawan yang diubah dan tekan *enter* untuk melakukan *request update* data karyawan untuk melakukan *update table* di *database*. *Delete* data karyawan dilakukan dengan cara menekan tombol *delete* yang akan melakukan *request delete* data karyawan untuk melakukan *delete* data karyawan di *database*.

2. Data Karyawan – Tambah Data

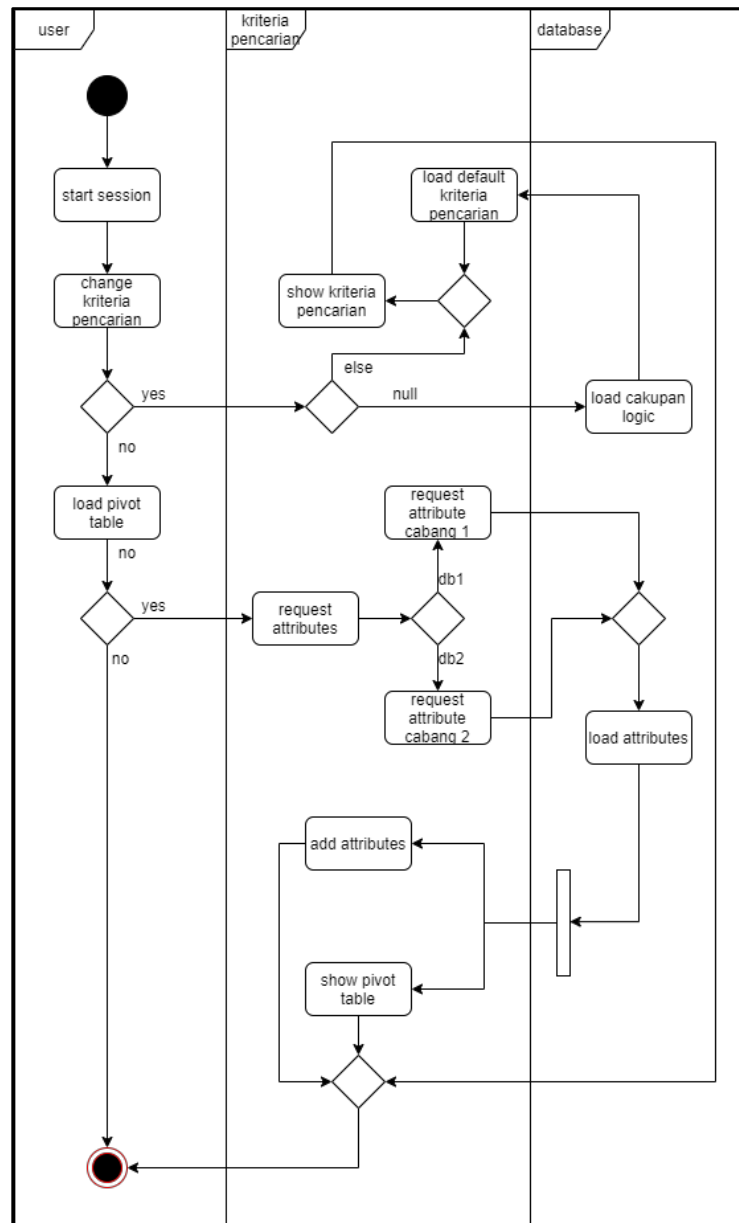


Gambar 3.6 *activity diagram* data karyawan – tambah data

Gambar 3.6 merupakan *activity diagram* data karyawan yang akan melakukan penambahan data karyawan. Penambahan data karyawan dimulai dari *input user* dan *user* dapat melakukan *reset* seluruh *input field* dan kemudian memasukkan data karyawan kembali.

Sistem akan melakukan *request* tambah data karyawan ke *database* dan melakukan penambahan data karyawan setelah *user* selesai melakukan *input*. *Database* akan mengembalikan seluruh datanya ke sistem untuk ditampilkan ke *user* setelah *inputan* data pada *database* selesai dijalankan.

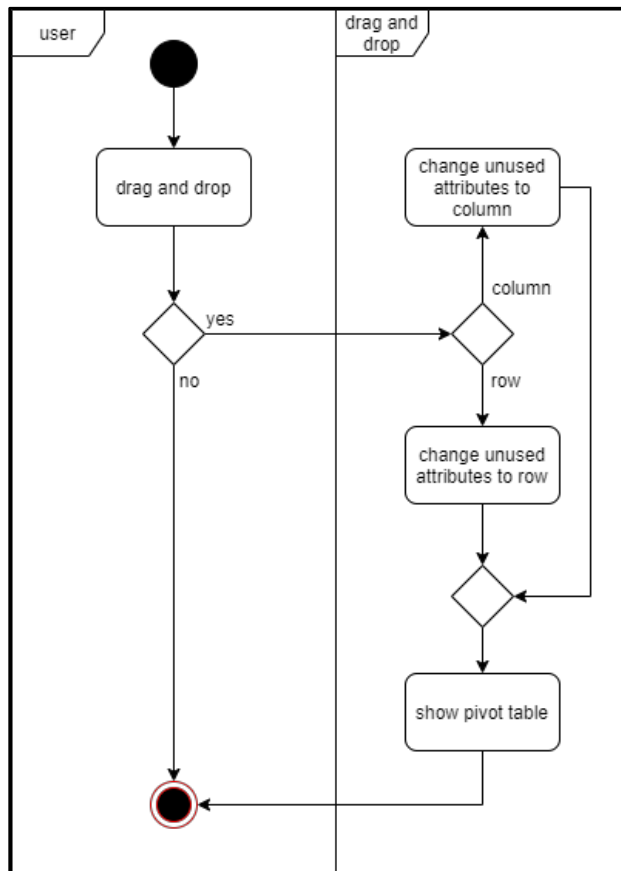
B.3.2. Activity Diagram Pivot Table



Gambar 3.7 activity diagram pivot table – kriteria pencarian

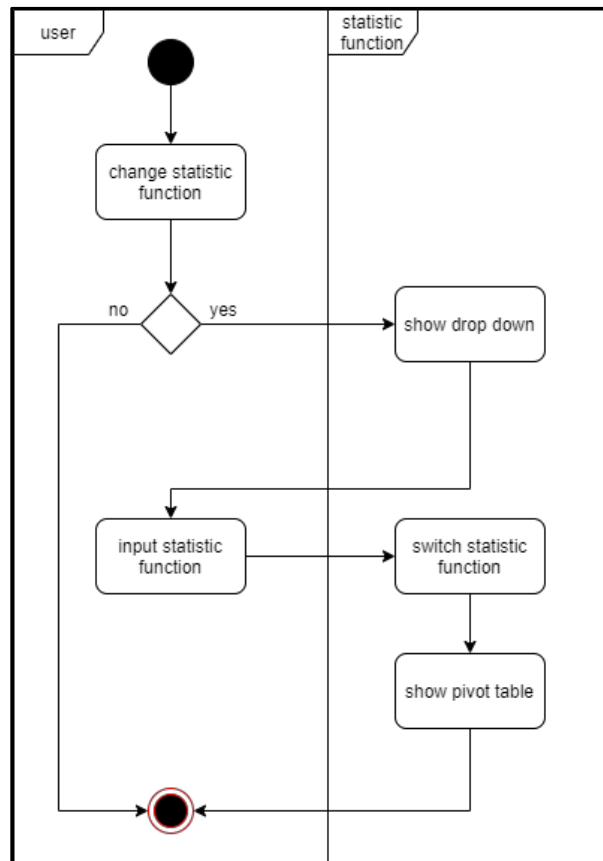
Gambar 3.7 merupakan *activity diagram pivot table* yang memiliki fungsi melakukan kriteria pencarian, *defaultnya* dan menampilkan *pivot table*. Kriteria pencarian dapat diubah oleh *user* dan apabila cakupan *logic* bernilai *null* atau belum memiliki *default* maka aplikasi akan melakukan pengisian cakupan *logic* dengan melakukan request ke *database* yang kemudian sistem akan membuat seluruh *default* dari kriteria pencarian dan menampilkannya ke *user*.

Tampilan *pivot table* dapat dimunculkan dengan melakukan *request attributes* ke *database* dan *database* akan melakukan *load attribute*. *Request attributes* dibagi menjadi 2 berdasarkan cabang yang digunakan karena memiliki *attribute* yang berbeda. Sistem akan menampilkan *pivot table* beserta seluruh *attribute* pada cabang yang dipilih setelah *load attribute* selesai dijalankan.



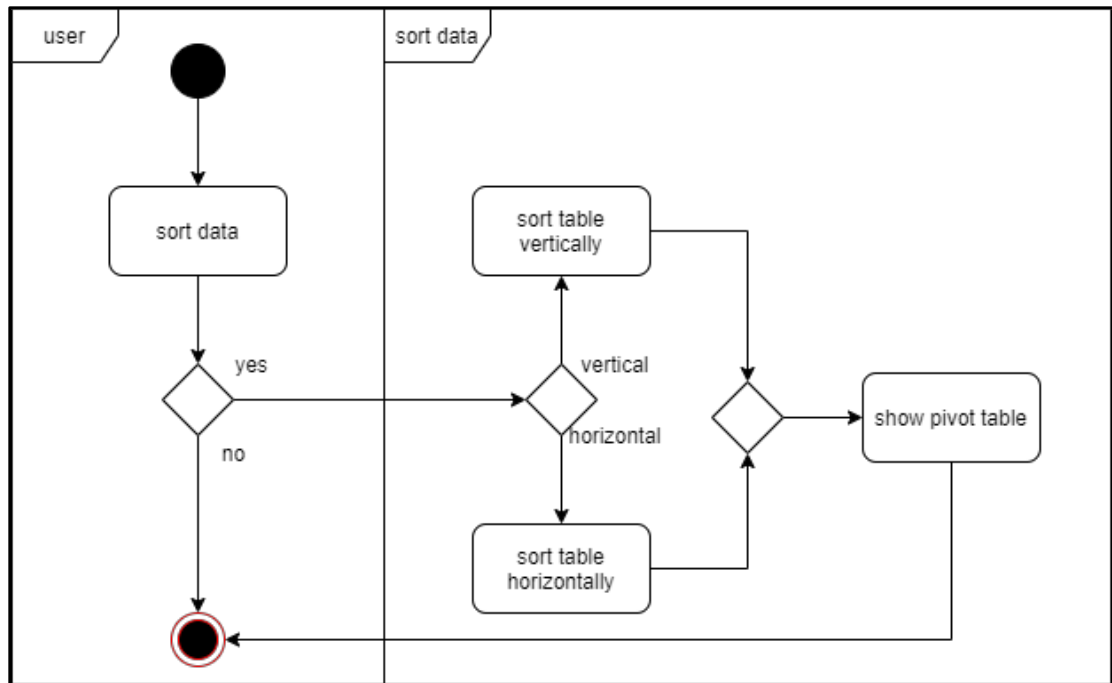
Gambar 3.8 *activity diagram pivot table – drag and drop*

Gambar 3.8 merupakan *activity diagram pivot table* yang memiliki fungsi untuk memindahkan *attribute* dari *unused attribute* ke kolom atau baris. Pemindahan *attribute* dapat dilakukan dengan cara melakukan *drag and drop* pada *attribute* ke kolom atau baris. Sistem akan menampilkan seluruh *value* pada *attribute* secara *horizontal* apabila *attribute* berada di baris dan secara *vertical* apabila *attribute* berada di kolom dan dilakukan perhitungan antara baris dan kolomnya.



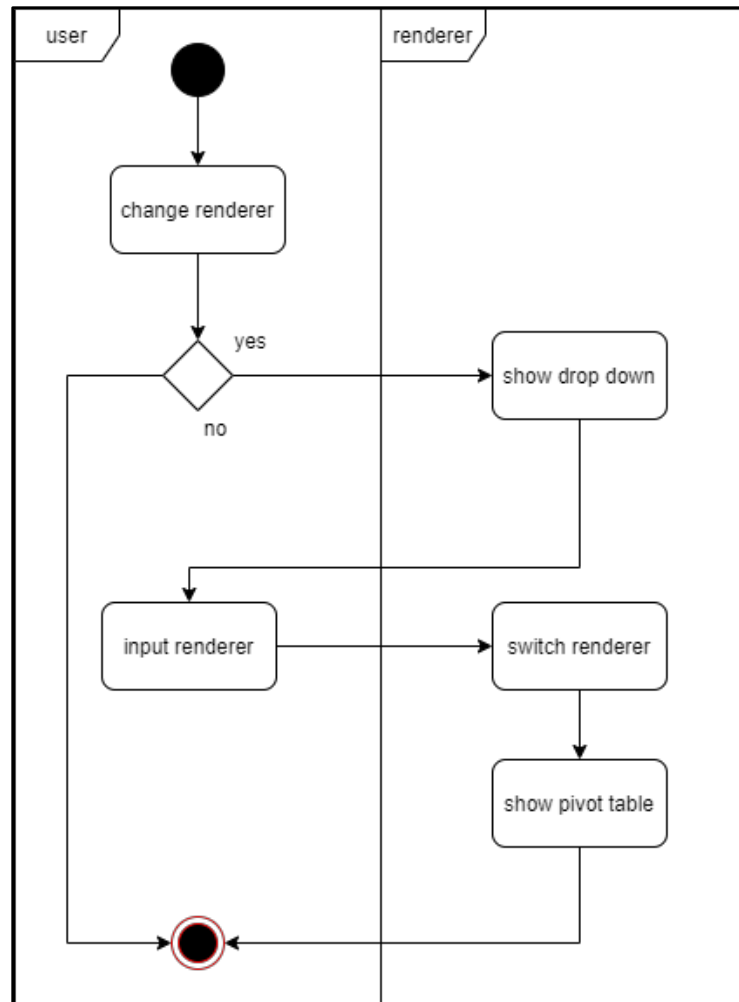
Gambar 3.9 *activity diagram pivot table – change statistic function*

Gambar 3.9 merupakan *activity diagram pivot table* yang berfungsi untuk mengubah teknik perhitungan yang digunakan pada tabel *pivot table*. *Drop down* yang berisi banyak *statistic function* akan dikeluarkan oleh sistem apabila *user* memilih *statistic function* dan *user* dapat memilih *statistic function* apa saja untuk mengubah *statistic function* dan *value* pada tabel akan berubah.



Gambar 3.10 *activity diagram pivot table – sort data*

Gambar 3.10 merupakan *activity diagram pivot table* yang berfungsi untuk melakukan *sorting* data. *User* dapat memilih 2 tipe *sorting* yakni, *vertical* dan *horizontal*. *Sorting* secara *vertical* akan mengubah urutan *value* pada kolom yang awalnya *ascending* menjadi *descending* dan berlaku kebalikan. *Sorting* secara *horizontal* akan mengubah urutan *value* pada baris yang awalnya *ascending* menjadi *descending* dan berlaku kebalikannya.



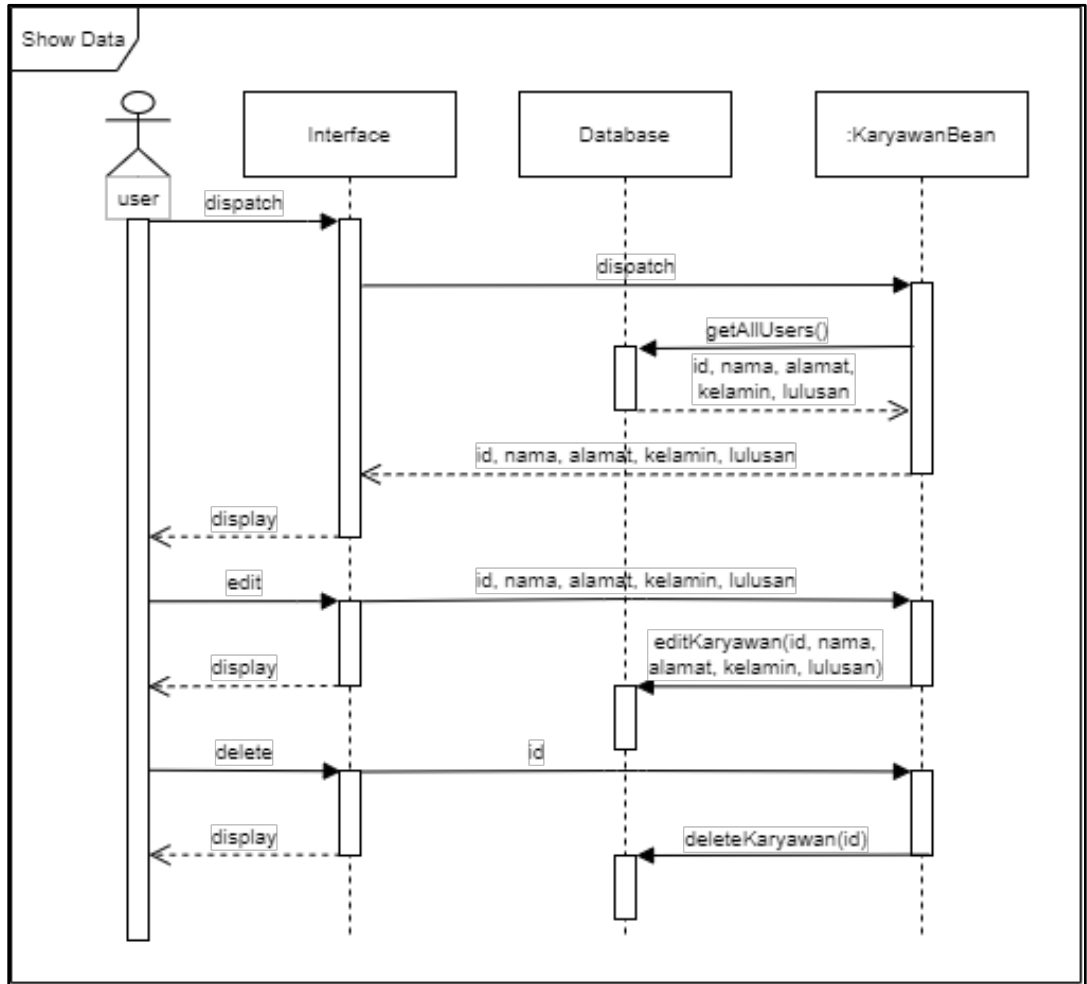
Gambar 3.11 *activity diagram pivot table – change renderer*

Gambar 3.11 merupakan *activity diagram pivot table* yang berfungsi untuk mengubah *renderer* dari *pivot table*. *Drop down* akan muncul ketika user menekan *renderer* dan *user* dapat memilih *renderer renderer* yang diinginkan untuk mengubah *renderer*. *Renderer* yang diubah akan menampilkan *pivot table* dalam bentuk yang berbeda.

B.4. Sequence Diagram

Sequence Diagram adalah perancangan *flow* perpindahan data yang digunakan pada sistem *pivot table* dan data karyawan.

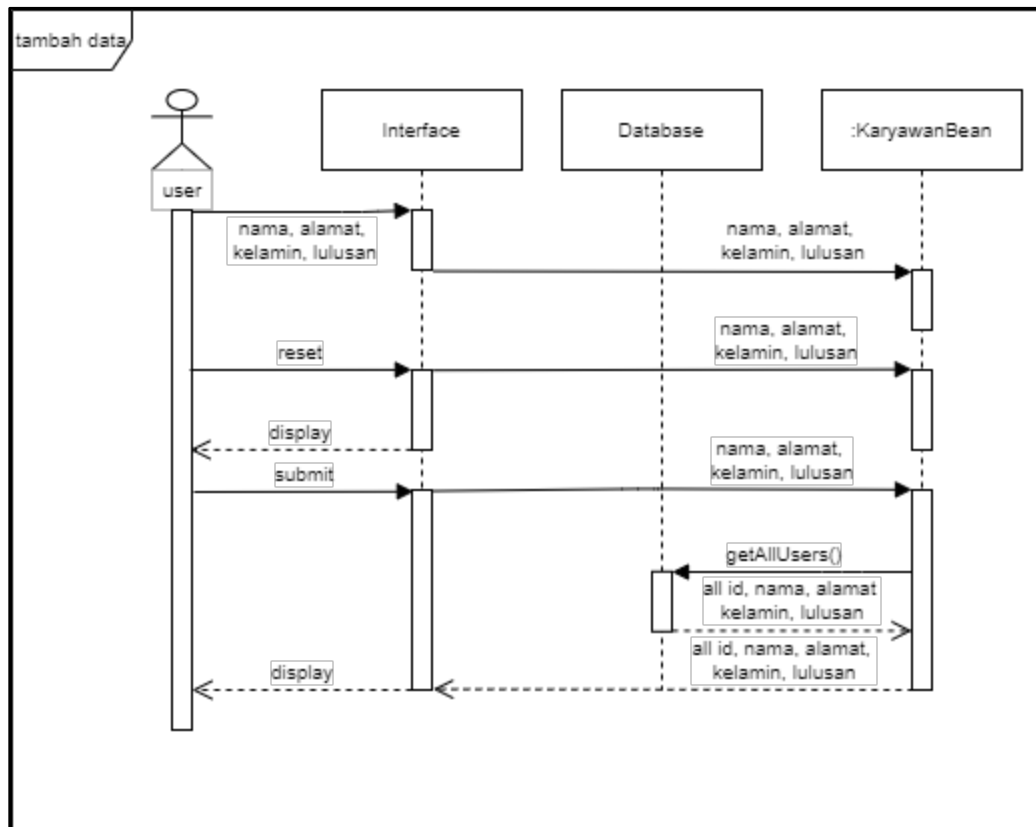
B.4.1. Sequence Diagram Data Karyawan



Gambar 3.12 *sequence diagram* data karyawan – *show data*

Gambar 3.12 merupakan *sequence diagram* data karyawan untuk menampilkan seluruh data karyawan, hapus data karyawan, dan *edit* data karyawan. Seluruh data karyawan akan ditampilkan ketika *user* pertama kali menjalankan aplikasi dimana *class KaryawanBean* akan melakukan *request* ke *database* untuk menerima seluruh data karyawan. *Database* akan mengembalikan seluruh *id*, nama, alamat, kelamin, dan lulusan dari data karyawan yang akan dikirim ke *interface* untuk ditampilkan.

Edit dilakukan dengan mengubah *object* pada *class* yang berisi *id*, nama, alamat, kelamin, dan lulusan dan melakukan request *editKaryawan* ke *database* dengan *value id*, nama, alamat, kelamin, dan lulusan. *Database* akan melakukan *update* nama, alamat, kelamin, dan lulusan berdasarkan *id* dari karyawan tersebut. *Delete* dilakukan dengan memanggil request *deleteKaryawan* yang berisi *id* karyawan. *Database* akan melakukan penghapusan data karyawan berdasarkan *id* karyawan tersebut.

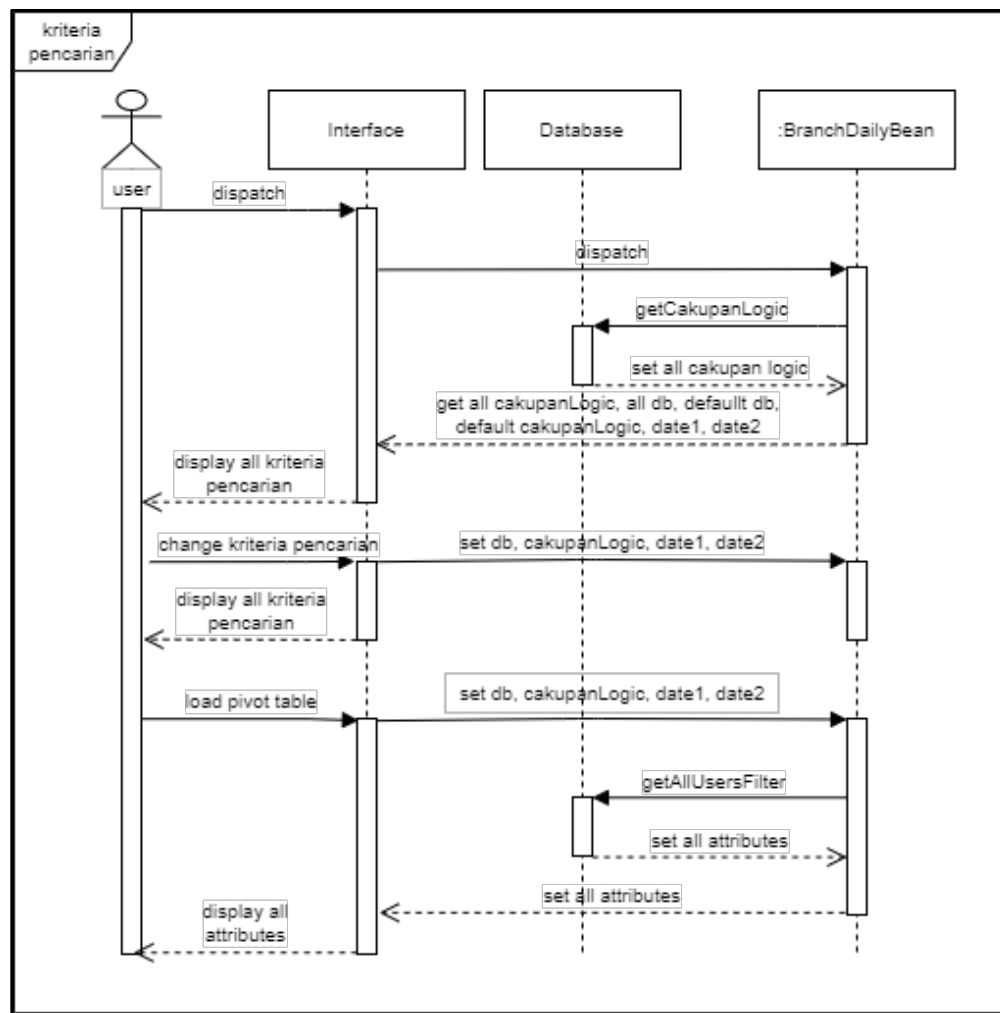


Gambar 3.13 *sequence diagram* data karyawan – tambah data

Gambar 3.14 merupakan *sequence diagram* data karyawan untuk menambah data karyawan. Data karyawan dapat ditambah dengan *user* melakukan input nama, alamat, kelamin, dan lulusan yang dikirim ke *class KaryawanBean*. *User* dapat membuang data yang sudah di-*input* dengan melakukan *reset* dan membersihkan *object* dari *class* dan membersihkan layar *user*.

Submit berfungsi untuk memasukkan data karyawan dengan mengirim ulang nama, alamat, kelamin, dan lulusan ke *class* dan melakukan *request addKaryawan* ke *database* dengan *value* nama, alamat, kelamin, dan lulusan. Tampilan *user* akan di-*update* untuk menampilkan seluruh data karyawan dengan cara *class* melakukan *request getAllUser* ke *database* setelah *addKaryawan* selesai berjalan. *Database* akan mengembalikan seluruh *id*, nama, alamat, kelamin, dan lulusan karyawan ke *object* untuk disimpan dan ditampilkan ke *user*.

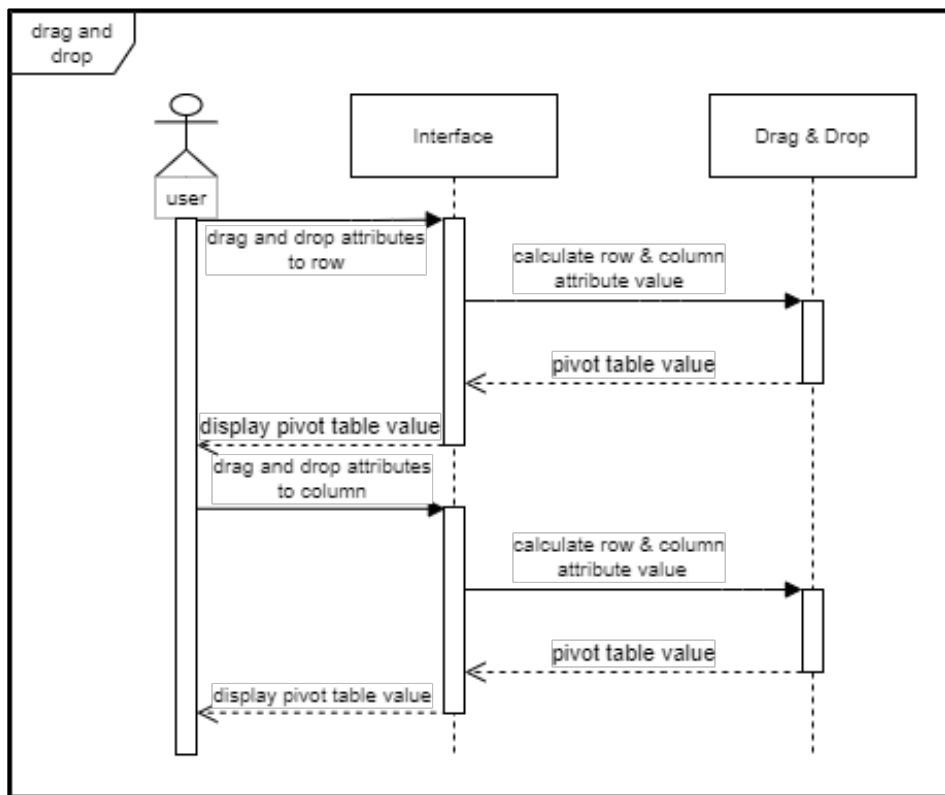
B.4.2. Sequence Diagram Pivot Table



Gambar 3.14 *sequence diagram pivot table – kriteria pencarian*

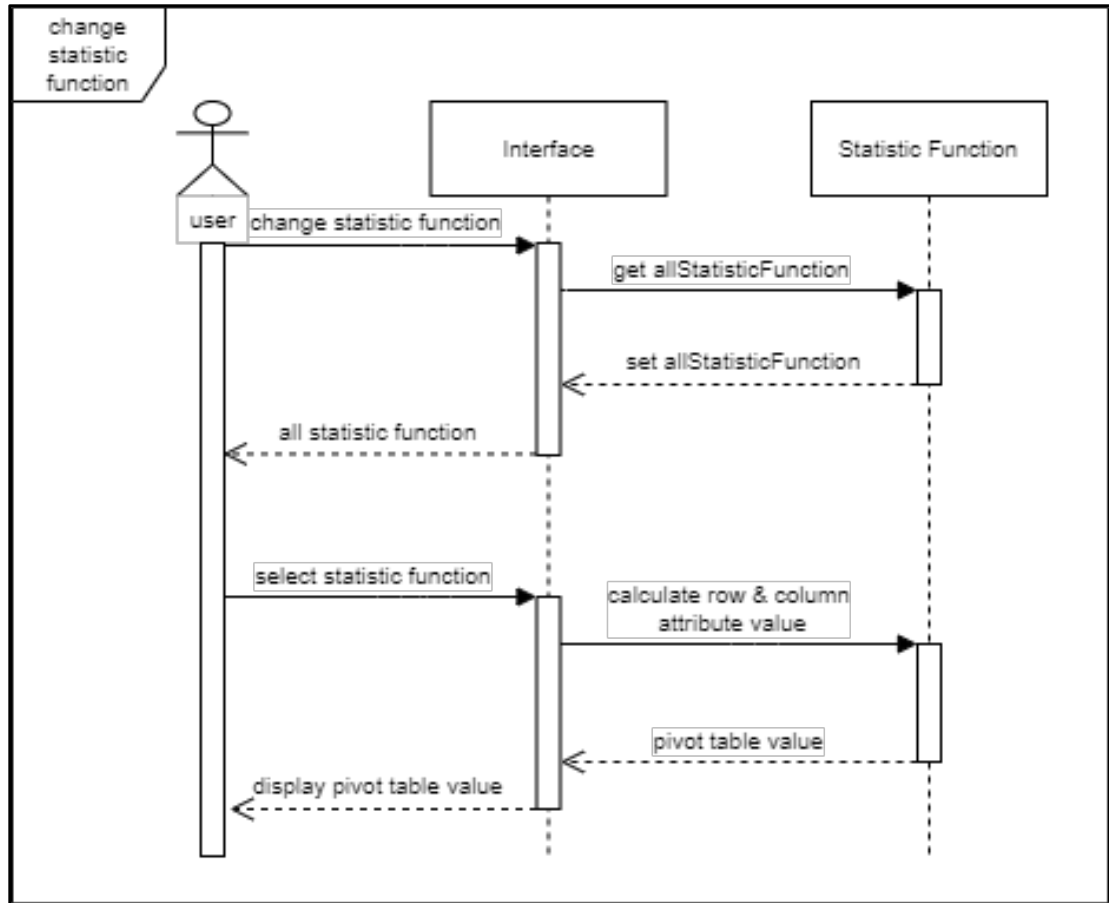
Gambar 3.14 merupakan *sequence diagram pivot table* yang berfungsi untuk mengubah kriteria pencarian, menampilkan *default* kriteria pencarian, dan menampilkan *pivot table*. *getCl()* adalah fungsi yang dijalankan oleh class *BranchDailyBean* untuk melakukan *request ke database* ketika *user* pertama kali menjalankan sistem *Database akan* mengirim seluruh cakupan *logic* dan *class* akan mengirimkan seluruh cakupan *logic*, cabang, *date1*, dan *date2* sebagai *default* kriteria.

Penggantian kriteria pencarian dapat dilakukan dengan cara mengubah cabang, cakupan *logic*, *date1*, dan *date2* ke *class* yang kemudian dapat digunakan untuk menampilkan *pivot table*. Pencarian menggunakan *request getAllFikter()* *class* ke *database* yang akan mengembalikan seluruh *attribute* yang akan dimasukkan ke dalam *pivot table* untuk ditampilkan ke *user*.



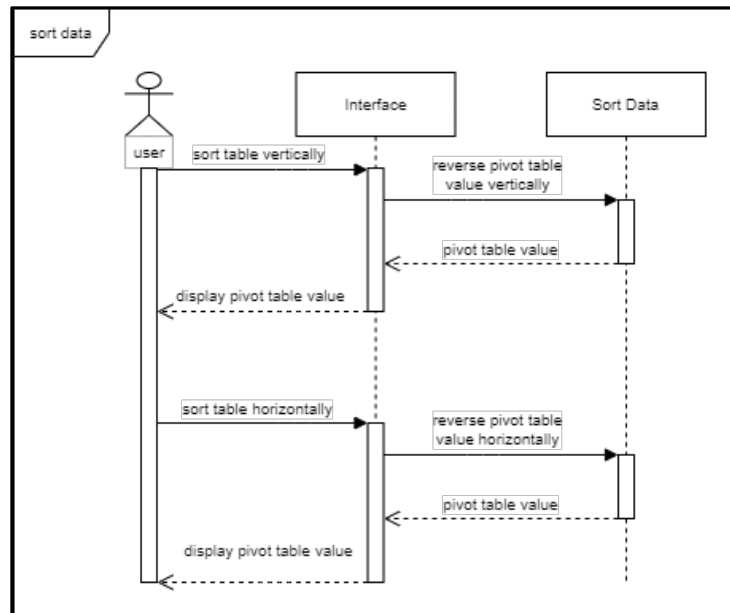
Gambar 3.15 *sequence diagram pivot table – drag and drop*

Gambar 3.15 merupakan *sequence diagram pivot table* yang bertujuan untuk menampilkan *value* pada *pivot table* dengan melakukan *drag and drop attribute* pada *unused attribute* ke baris atau kolom. *User* dapat langsung menarik *attribute* yang berada pada *unused attribute* ke baris atau kolom dan sistem akan melakukan perhitungan antara baris dan kolomnya sesuai dengan *statistic function* yang berlaku.



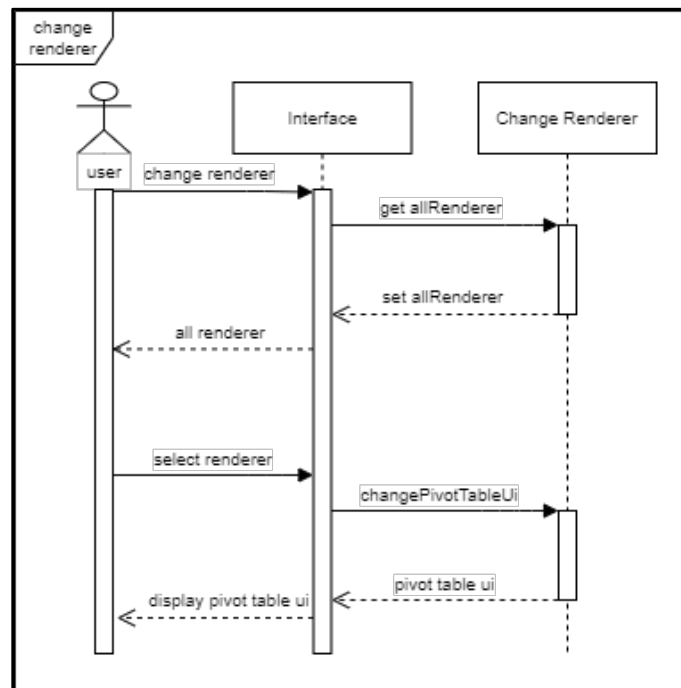
Gambar 3.16 *sequence diagram pivot table – change statistic function*

Gambar 3.16 merupakan *sequence diagram pivot table* yang berfungsi untuk mengubah *statistic function* dengan *user* mengganti *statistic function* dan sistem akan mengembalikan *drop down* seluruh *statistic function*. *User* diminta untuk memilih satu *statistic function* dan sistem akan menampilkan *value* baru pada *pivot table* dengan melakukan kalkulasi antara baris dan kolom menggunakan *statistic function* yang baru.



Gambar 3.17 *sequence diagram pivot table – sort data*

Gambar 3.17 merupakan *sequence diagram pivot table* yang akan melakukan *sorting value* pada *pivot table*. *Sorting* dapat dilakukan secara *vertical* atau *horizontal* yang keduanya akan mengembalikan *value* dalam *order* yang dibalik secara *vertical* atau *horizaontal*.



Gambar 3.18 *sequence diagram pivot table – change renderer*

Gambar 3.18 merupakan *sequence diagram pivot table* yang berfungsi untuk mengubah *renderer* dengan cara *user* mengubah *renderer* dan sistem akan mengembalikan seluruh *renderer* yang tersedia dalam bentuk *drop down*. *User* diminta untuk memilih satu *renderer* yang akan digunakan dan sistem akan mengubah tampilan *pivot table* sesuai dengan *renderer* yang dipilih *user*.







B.5. Rancangan Antarmuka

Rancangan Antarmuka merupakan sebuah *mockup* yang dibuat dan akan diajukan berdasarkan apa yang diminta oleh pembimbing lapangan. Rancangan antarmuka ini dibuat dengan menggunakan word dan excel.

B.5.1. Rancangan Antarmuka Data Karyawan

1. Halaman Awal

Gambar 3.19 menunjukkan halaman halaman awal dari aplikasi data karyawan. Halaman ini menunjukkan *id*, nama, alamat, jenis kelamin, dan lulusan seluruh karyawan yang terdaftar. Terdapat *button edit* yang berfungsi untuk mengubah data karyawan dan *button delete* untuk menghapus data karyawan.

DATA KARYAWAN					+
Id	Nama	Alamat	Jenis Kelamin	Lulusan	
1	X	Y	L	SD	 
2	X	Y	P	SMP	 
3	X	Y	L	SMA	 

Gambar 3.19 Data Karyawan *mockup* – halaman awal

2. Halaman Tambah Data Karyawan

Gambar 3.20 menunjukkan halaman tambah data karyawan dari aplikasi data karyawan. Halaman ini meminta *user* untuk meng-*input* nama, alamat, jenis kelamin, dan lulusan karyawan. Tidak ada *default* yang dijalankan pada setiap *input field*. Ketika tombol *reset* ditekan, seluruh *input field* akan dibersihkan. Ketika tombol *submit* ditekan halaman akan berpindah ke halaman awal.

The image shows a web form titled "DATA KARYAWAN". It contains four input fields stacked vertically, each with a label to its left: "Nama", "Alamat", "Jenis Kelamin", and "Lulusan". The labels "Jenis Kelamin" and "Lulusan" are underlined. Below the input fields are two buttons: "Reset" and "Submit".

Gambar 3.20 Data Karyawan *mockup* – halaman tambah

B.5.2. Rancangan Antarmuka Pivot Table

1. Halaman Utama

Gambar 3.21 menunjukkan halaman utama dari sistem *pivot table*. Halaman ini terdiri dari 2 bagian yakni, kriteria pencarian dan *pivot table*. *Default* kriteria pencarian akan dijalankan ketika pertama kali sistem dijalankan sehingga *user* tidak perlu melakukan *input* dan default dari *pivot table* adalah *unused attributes*, *baris*, dan *kolomnya masih kosong*. *unused attributes* berada pada kotak yang memiliki nama, *age*, dan *gender*. *Baris* berada pada kotak yang memiliki *province* dan *kolom* berada pada kotak yang memiliki *party*.

> Kriteria Pencarian

Analisa Data: Cabang v Cakupan Logic: RO Reguler v Panjang Data: 30 hari 60 hari 90 hari Periode: 1 April 2019 # - 1 Juli 2019 # Tampilkan

Table: Name v Age v Gender v

Count: Party v

Province	Party	Bloc Quebecois	Conservative	Green	Liberal	NDP	Totals
Alberta			27			1	28
British Columbia			21	1	2	12	36
Manitoba			11		1	2	14
New Brunswick			8		1	1	10
Newfoundland and Labrador			1		4	2	7
Nova Scotia			4		4	3	11
Ontario			73		11	22	106
Prince Edward Island			1		3		4
Quebec		4	5		7	59	75
Saskatchewan			13		1		14
Territories			2			1	3
Totals		4	166	1	34	103	308







Gambar 3.21 *Pivot Table mockup* – halaman utama

C. Implementasi

Implementasi adalah hasil dari rancangan antar muka yang sudah diaplikasikan ke sistem data karyawan dan *pivot table*.

C.1. Implementasi Aplikasi Data Karyawan







Pada Gambar 3.22, tekan *button edit* atau tombol yang berbentuk pensil untuk mengubah data. Tekan *button delete* atau tombol yang berbentuk tong sampah untuk membuang data. Tekan *button tambah* untuk melakukan penambahan data.

DATA KARYAWAN					+
Id	Nama	Alamat	Jenis Kelamin	Lulusan	
1	Budi	Duta	L	SD	 
2	Judi	Bandara	L	SMP	 
3	Dedi	Mc D	L	SMA	 

Gambar 3.22 Halaman Awal Data Karyawan





Tekan *button edit* pada Gambar 3.22 untuk menampilkan *drop down* jenis kelamin dan lulusan. Jenis kelamin memiliki *drop down* L dan P yang dimana L adalah laki-laki dan P adalah perempuan.

Lulusan memiliki *drop down* TK, SD, SMP, SMA, SMK, S1, S2, dan S3. Nama dan alamat dapat langsung diubah dengan menekan *field* pada bagian nama atau alamat. Tekan *enter* atau klik dibagian mana saja untuk melakukan *update* seperti Gambar 3.23.

DATA KARYAWAN					+
Id	Nama	Alamat	Jenis Kelamin	Lulusan	
1	Budi	Duta	L	SD	 
2	Judi	Bandara	L <input type="text" value="v"/>	SMP <input type="text" value="v"/>	 
3	Dedi	Mc D	L	SMA	 

Gambar 3.23 Halaman Awal Data Karyawan - Edit

Tekan *button delete* pada Gambar 3.22 untuk membuang satu baris *table*. Database dan object akan melakukan pembuangan data dengan menggunakan *id* dan apabila dilakukan penambahan data, *id* akan melanjutkan *id* terakhir dibuat ditambah 1 seperti Gambar 3.24.

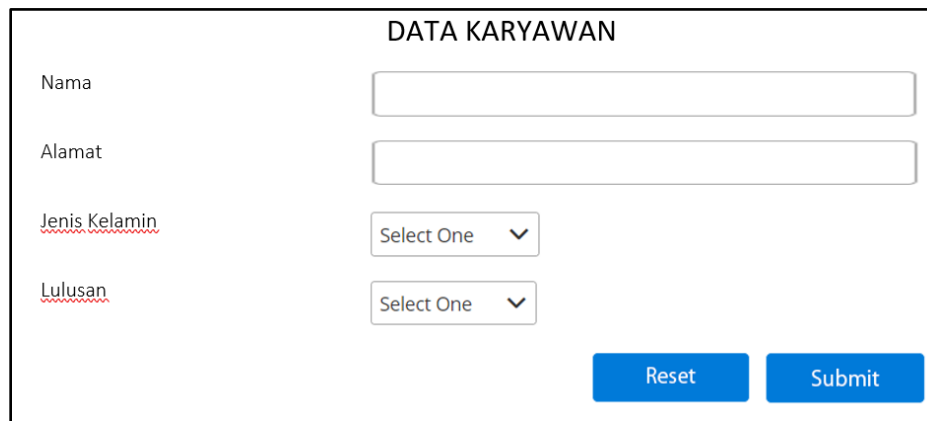
DATA KARYAWAN					+
Id	Nama	Alamat	Jenis Kelamin	Lulusan	
1	Budi	Duta	L	SD	 
3	Dedi	Mc D	L <input type="text" value="v"/>	SMA	 

Gambar 3.24 Halaman Awal Data Karyawan - Delete

Tekan *button* tambah pada gambar 3.22 untuk berpindah ke halaman tambah data karyawan. Pada halaman tambah data karyawan, terdapat nama dan alamat yang memiliki tipe *inputtext* dan jenis kelamin dan lulusan memiliki tipe *selectonemenu*.

Seperti pada halaman awal data karyawan, jenis kelamin memiliki 2 *values* yakni L dan P. Lulusan memiliki *values* TK, SD, SMP, SMA, SMK, S1, S2, dan S3. *button reset* akan meng-*update grid* yang digunakan untuk menampung tampilan layer dengan melakukan *load* ulang tampilan atau *reset* tampilan.

Button submit akan melakukan *submit* yang akan melakukan *update* pada *object* pada data karyawan dan melakukan *update* database dengan menambah *id* menggunakan *auto increment*. Setelah *update* dilakukan, aplikasi akan berpindah ke halaman awal dengan data yang sudah ter-*update* seperti pada Gambar 3.25.

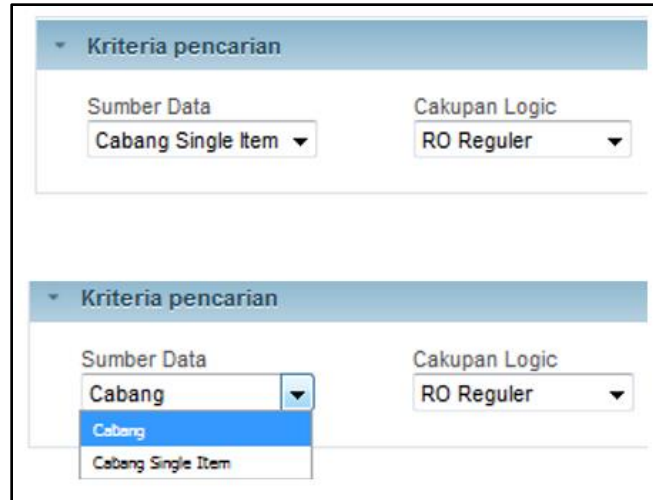


The image shows a web form titled "DATA KARYAWAN". It has four input fields: "Nama" (text), "Alamat" (text), "Jenis Kelamin" (dropdown menu with "Select One" and a downward arrow), and "Lulusan" (dropdown menu with "Select One" and a downward arrow). At the bottom right, there are two blue buttons labeled "Reset" and "Submit".

Gambar 3.25 Halaman Tambah Data Karyawan

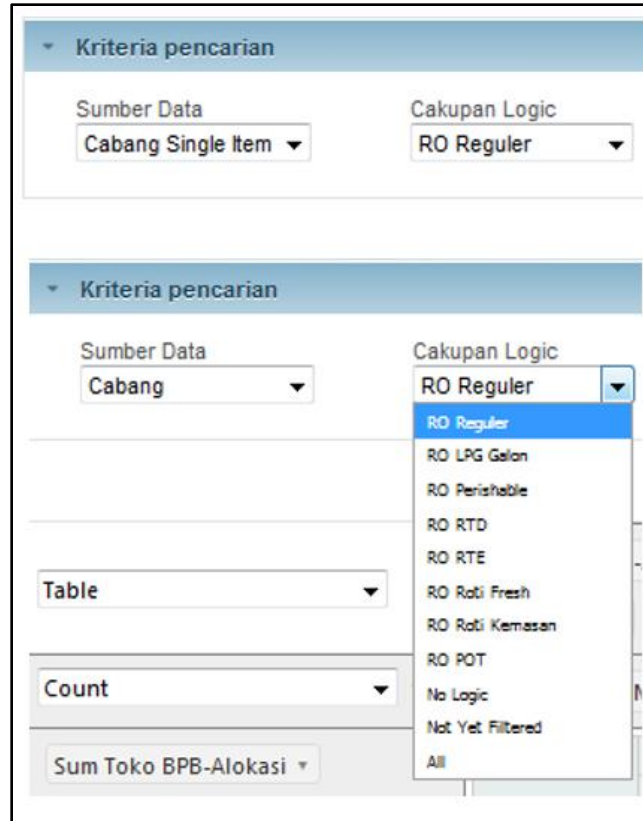
C.2. Implementasi Sistem Pivot Table

Pemilihan sumber data menggunakan tipe *select one menu* yang memiliki *default* sumber data cabang. Pemilihan sumber data dapat dilakukan dengan cara menekan di bagian putih mana saja sumber data dan panah bawah dari sumber *data*. Sumber data akan memberikan *drop down* yang berisikan toko-toko yang terdaftar di *database* setelah sumber data ditekan dan tekan toko yang diinginkan untuk ditampilkan ke *pivot table* seperti pada Gambar 3.26.



Gambar 3.26 Kriteria Pencarian - Sumber Data

Pemilihan cakupan *logic* menggunakan *select one menu* seperti sumber *data* yang memiliki *default* RO Reguler. Pemilihan cakupan *logic* dapat dilakukan dengan cara menekan di bagian putih mana saja cakupan *logic* dan panah bawah dari cakupan *logic*. Setelah ditekan, cakupan *logic* akan memberikan *drop down* yang berisikan seluruh cakupan *logic* yang terdaftar di *database* pada toko yang dipilih dan tekan cakupan *logic* yang diinginkan untuk ditampilkan ke *pivot table* seperti pada Gambar 3.27.



Gambar 3.27 Kriteria Pencarian - Cakupan *logic*

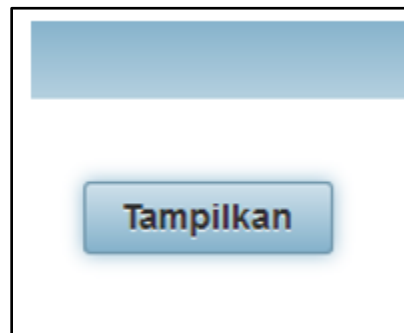
Periode memiliki tipe data *calendar* yang *default*-nya adalah *current date* atau tanggal hari ini. Terdapat 2 cara untuk memilih periode. Cara pertama, tekan button kalender untuk memunculkan kalender yang kemudian dapat dipilih. Cara kedua, tekan bagian putih periode dibagian kanan dan mengubah tanggal sesuai yang kita inginkan mengikuti *default* dari periode.

Panjang data memiliki tipe *radio button* dan *default* 30 hari. Tekan lingkaran putih disebelah pilihan hari untuk memilih Panjang data dan periode awal akan berubah mengikuti panjang data seperti Gambar 3.28.

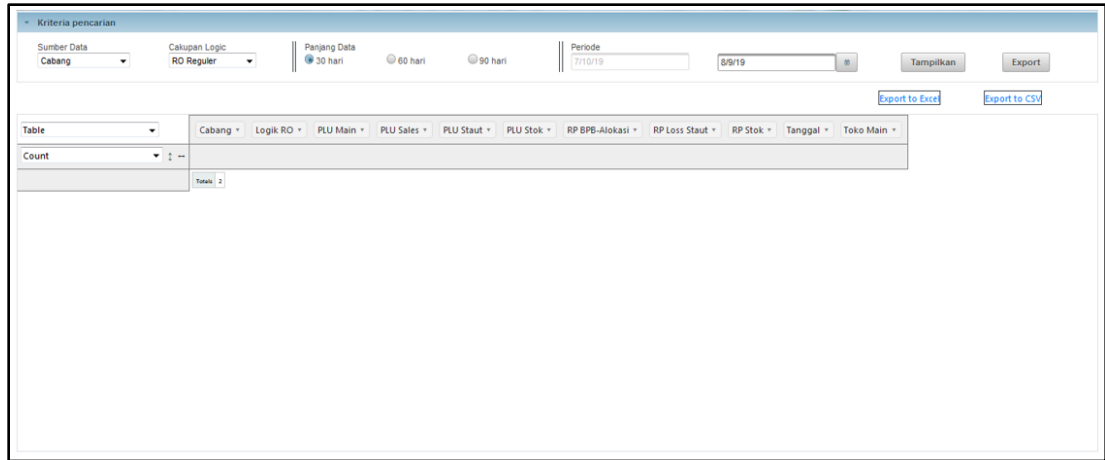
The image shows a search criteria form with two main sections. The first section is titled 'Panjang Data' and contains three radio button options: '30 hari' (which is selected), '60 hari', and '90 hari'. The second section is titled 'Periode' and contains two date input fields. The first field contains the date '6/25/19' and the second field contains '7/25/19'. To the right of the second date field is a small square button with a calendar icon.

Gambar 3.28 Kriteria Pencarian - Panjang data & periode

Pada gambar 3.29, terdapat *button* tampilkan yang berfungsi untuk menjalankan *query* ke *database* setelah selesai melakukan *step* diatas dan *attributes* akan ditambahkan ke pivot table seperti pada Gambar 3.30.

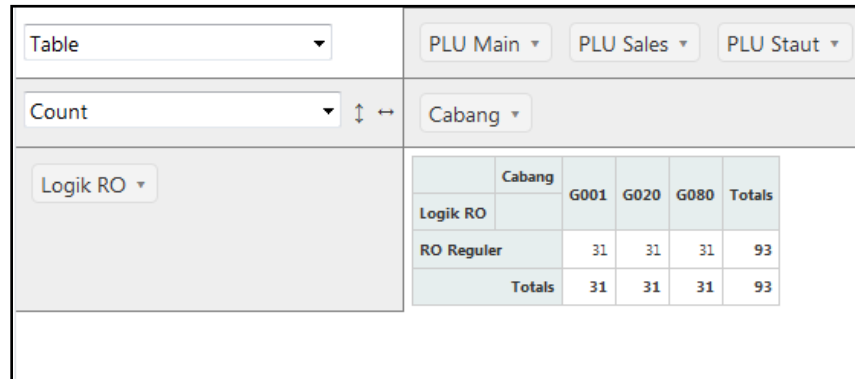


Gambar 3.29 Kriteria Pencarian - *Button* tampilkan



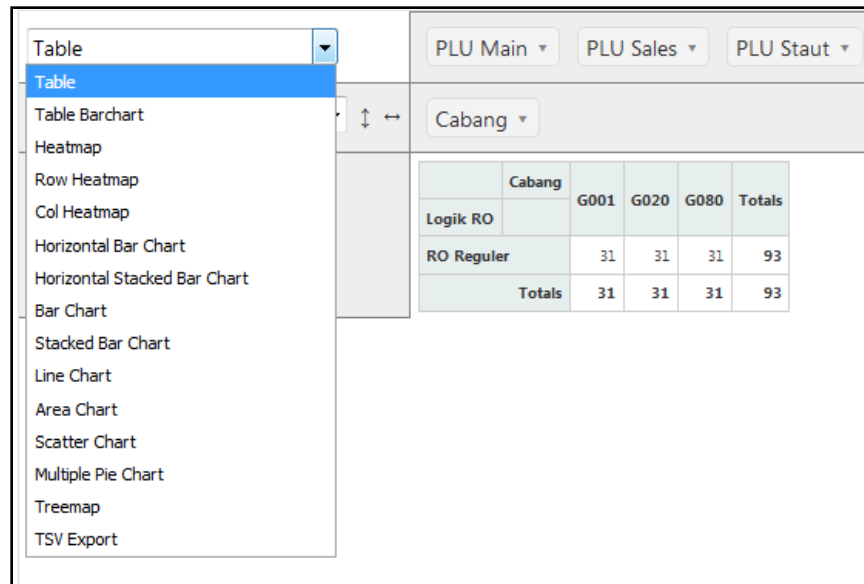
Gambar 3.30 *Pivot Table*

Pindahkan attributes ke baris dan kolom untuk menampilkan data *pivot table*. Untuk memindahkan *attributes* dapat dilakukan *drag and drop* yang tersedia pada *unused attributes* seperti pada Gambar 3.31.



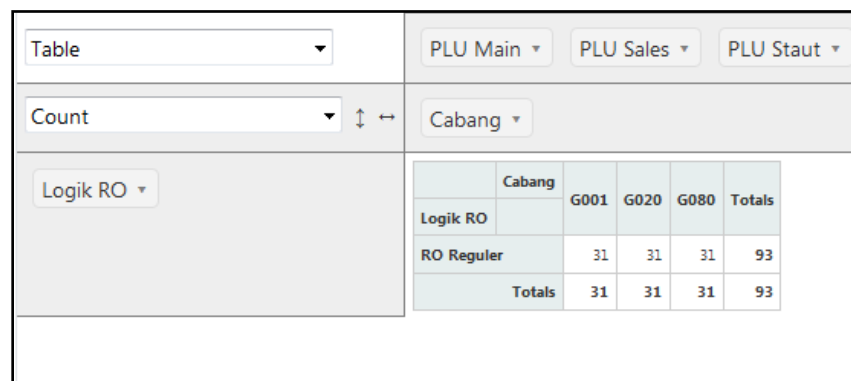
Gambar 3.31 *Attributes*

Default dari *renderer* adalah *table*. Tekan bagian putih mana saja *renderer* atau tekan panah bawah dari *renderer* untuk mengubah *renderer*. *Renderer* akan memunculkan *drop down* yang berisi berbagai macam *renderer* apabila *field renderer* ditekan dan tekan *renderer* yang diinginkan untuk mengubah *renderer* seperti Gambar 3.32.



Gambar 3.32 *Renderer*

Default dari *statistic function* adalah *count*. Tekan bagian putih mana saja *statistic function* atau tekan panah bawah dari *statistic function* untuk mengubah *statistic function*. akan muncul *Drop down* yang berisi berbagai macam *statistic* dan tekan *statistic function* yang diinginkan untuk mengubah *statistic function* akan muncul setelah *statistic function* ditekan seperti Gambar 3.33.



Gambar 3.33 *Statistic Function*

Gambar 3.34 menunjukkan fungsi untuk melakukan *export table* dari *pivot table* menjadi *xlsx* atau *csv*. Tombol *export* berwarna abu-abu akan mengubah *pivot table*

table ke dalam bentuk csv dan tidak membutuhkan lokasi *download* karena lokasinya sudah *fix*. Tombol *export to excel* dan *export to csv* akan mengubah *pivot table* ke dalam bentuk excel atau csv dan akan menampilkan *modal download* seperti melakukan *download* biasanya pada berbagai macam *browser*.



Gambar 3.34 *Function Export*

3.3.2. Kendala yang Ditemukan

Beberapa kendala yang ditemukan saat pelaksanaan kerja magang adalah sebagai berikut.

1. Terbatasnya pengetahuan yang diakibatkan oleh penggunaan Bahasa pemrograman yang baru digunakan dan belum dikuasai secara maksimal.
2. Kurangnya ilmu pengetahuan tentang apa itu *pivot table* dan bagaimana cara *pivot table* bekerja yang diakibatkan karena belum memiliki ilmu *data science* yang cukup.
3. Munculnya istilah-istilah baru yang belum pernah digunakan dalam dunia perkuliahan seperti *user* yang berarti perusahaan melaksanakan magang dan *client* yang berarti *manager* dari perusahaan tempat pelaksanaan magang.

3.3.3. Solusi atas Kendala yang Ditemukan

Solusi atas kendala yang ditemukan saat pelaksanaan kerja magang adalah sebagai berikut.

1. Melakukan *research* pribadi mengenai bahasa pemrograman yang digunakan secara online dan mengikuti program pembelajaran dari perusahaan.
2. Sering-sering berkoordinasi, bertanya, dan mendengarkan dengan baik instruksi dan masukan dari pembimbing lapangan mengenai *pivot table* dan bagian dari sistem yang tidak dimengerti.
3. Sering-sering berbicara dan bertanya dengan teman kantor untuk memperluas istilah-istilah yang digunakan di perkantoran.