**Analisis objek orientasi programing**

**Sistem aplikasi kepegawaian**



**OLEH:**

**Nama : sri devi**

**Stambuk : 21916051**

**Kelas : 4 B PTI**

**PROGRAM STUDY PENDIDIKAN TEKNOLOGI INFORMASI**

**FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN**

**UNIVERSITAS MUHAMMADIYAH KENDARI**

**KENDARI**

**2021**

**Alat Pendukung Pembuatan Sistem**

Alat pendukung pembuatan Sistem Informasi Kepegawaian terdiri dari beberapa perangkat lunak dan perangkat keras. Adapun alat pendukung pembuatan sistem yaitu sebagai berikut :

**Perangkat Lunak (Software)**

Perangkat lunak yang dibutuhkan Sistem Informasi Kepegawaian

* Sistem Operasi : Windows 10 64bit dan Android Versi 5.0
* Database : MySQL.
* Tools : Android Studio, IntelliJ IDEA, Edraw Max, Xampp (MySQL), Ms. Word dan Notepad.

**Perangkat Keras (Hardware)**

Perangkat keras dengan rekomendasi minimum yang digunakan dalam pembuatan Sistem Informasi Kepegawaian adalah sebagai berikut :

* Komputer atau laptop, dengan spesifikasi minimal :

1. Processor Core i3.
2. Hardisk 500 GB.
3. RAM 4 GB.

* Smartphone Android.

**Analisis Permasalahan**

Permasalahan yang terjadi pada sistem yang sedang berjalan saat ini adalah sebagai berikut :

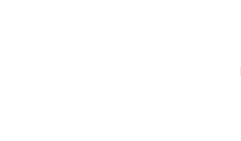
1. Data mutasi dan kenaikan pangkat pegawai Kejaksaan yang diolah menggunakan aplikasi Office tidak memiliki basis data sehingga tidak terdapat back up data. Jika terjadi kesalahan seperi “human error” (terjadi penghapusan data), maka data tersebut suli dikembalikan seperti semula.
2. Presensi pegawai dilakukan dengan cara mengisi buku presensi setiap hari pada jam masuk kerja. Proses tersebut dapat menimbulkan kecurangan pada pengisian data presensi pegawai jika ada pegawai yang “menitip” presensi(pemalsuan tanda tangan jika pegawai tersebut tidak hadir) kepada pegawai lain.

**Quick Plan**

Dari analisis permasalahan yang telah diuraikan sebelumnya, maka untuk mengurangi permasalahan yang terjadi pada Kejaksaan Negeri Lampung Timur pada penelitian ini peneliti mengusulkan suatu sistem informasi kepegawaian yang di dalamnya terdapat presensi pegawai menggunakan QR Code berbasis Android dan pengolahan data mutasi serta kenaikan pangkat pegawai berbasis Desktop.

**Modelling Quick Design**

Setelah didapat ide mengenai sistem yang akan dibuat, maka pada tahap ini peneliti membuat perancangan/pemodelan sistem. Adapun perancangan sistem diusulkan terdiri dari perancangan arsitektur sistem, use case diagram, activity diagram, class diagram dan interface sistem. Perancangan ini dibuat menggunakan aplikasi Edraw Max.



Data pegawai Data Mutasi

Data kenaikan pangkat Pencarian data & laporan

Kaur Kepegaw aian

Inf ormasi data mutasi, kenaikan pangkat, pegawai dan laporan

Data store

Network Cloud

Data store

Data hasil scan

Code

Scan code



**Keterangan** :

1. Kaur Kepegawaian selaku admin memasukkan data pegawai, data mutasi dan kenaikan pangkat ke dalam sistem dan data tersebut tersimpan ke dalam basis data. Ketika Kaur Kepegawaian membutuhkan data-data tersebut dan melakukan pencarian, maka data yang dibutuhkan akan ditampilkan. Ketika Kaur Kepegawaian membutuhkan laporan, maka informasi data laporan tersebut akan tampil sesuai dengan yang dibutuhkan.
2. Presensi pegawai dilakukan dengan cara men-scan code yang telah tampil di dalam komputer. Pegawai Kejaksaan men-scan code yang telah tampil di dalam komputer menggunakan smartphone Android. Data hasil scan tersebut tersimpan di dalam database.

**Implementasi Sistem**

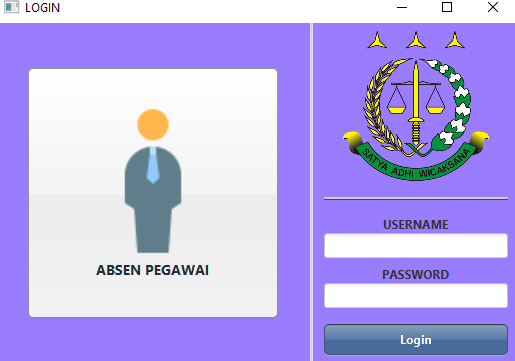
Implementasi dari sistem yang telah dirancang sebelumnya adalah terdiri dari 2 (dua) hak akses sistem, yaitu implementasi sistem berbasiskan desktop dengan hak akses Kaur Kepegawaian, dan implementasi berbasis Android yang diakses.

**Implementasi Sistem (Desktop) Akses Kaur Kepegawaian**

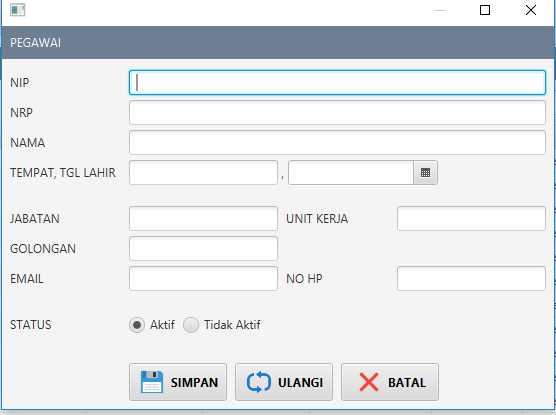
Implementasi dari sistem yang diusulkan berbasis desktop ini digunakan oleh Kaur Kepegawaian dalam pengolahan data pegawai, mutasi, kenaikan pangkat, melihat data presensi pegawai, dan laporan. Sistem ini dibuat dengan bahasa pemograman Java menggunakan aplikasi IntelliJ IDEA. Adapun implementasi dari sistem ini adalah sebagai berikut :

1. Implementasi Halaman Login Kaur Kepegawaian

Halaman ini digunakan untuk login Kaur Kepegawaian beserta akses absensi pegawai lainnya termasuk Kaur Kepegawaian menggunakan QR code.

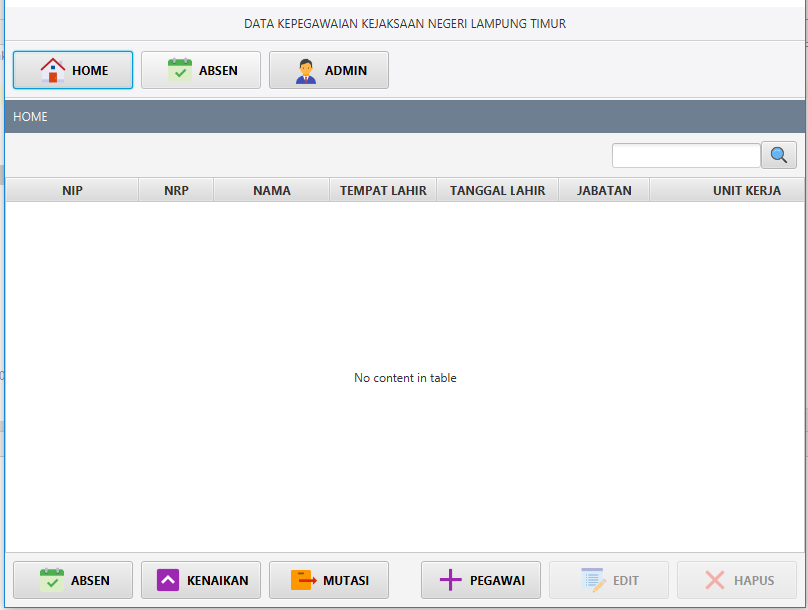


1. **Implementasi Halaman Input Pegawai**

Implementasi ini dipergunakan oleh Kaur Kepegawaian untuk dapat menambah data pegawai ke dalam sistem.

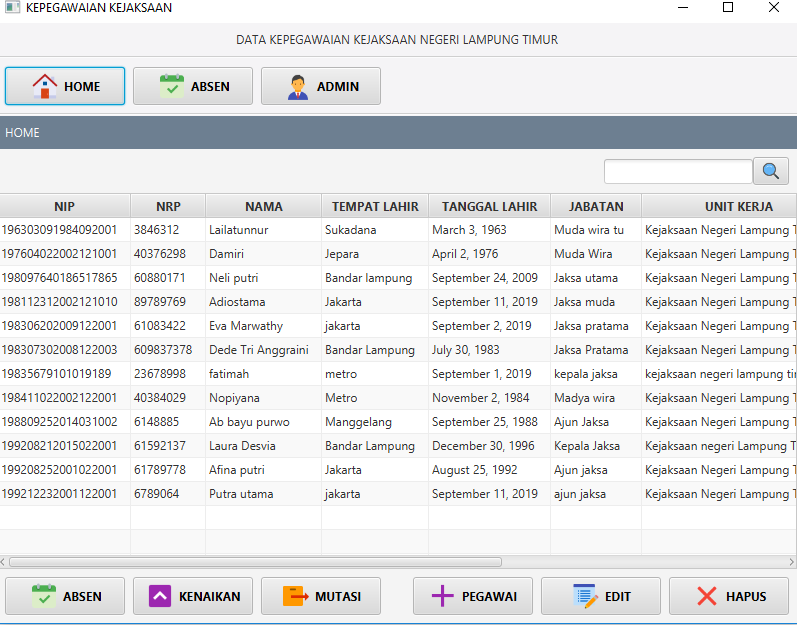
1. **Implementasi Proses Halaman Utama**

Implementasi halaman utama berisi menu-menu pendukung yang dapat digunakan oleh Kaur kepegawaian dalam mengolah data mutasi, data pegawai, data admin dan data kenaikan pangkat



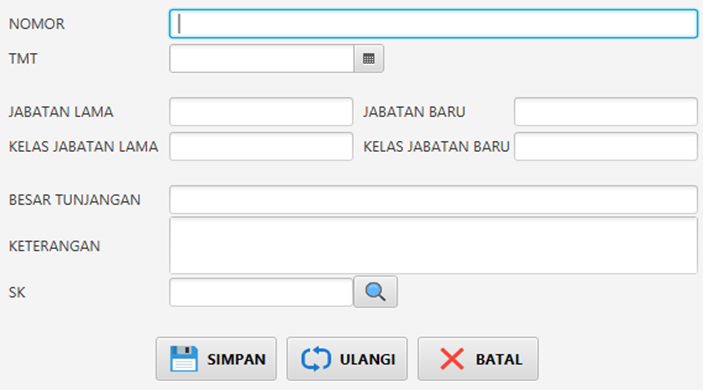
1. implementasi Output Data Kepegawaian

Implementasi ini hasil output dari data kepegawaian ke dalam sistem adalah seperti pada Gambar



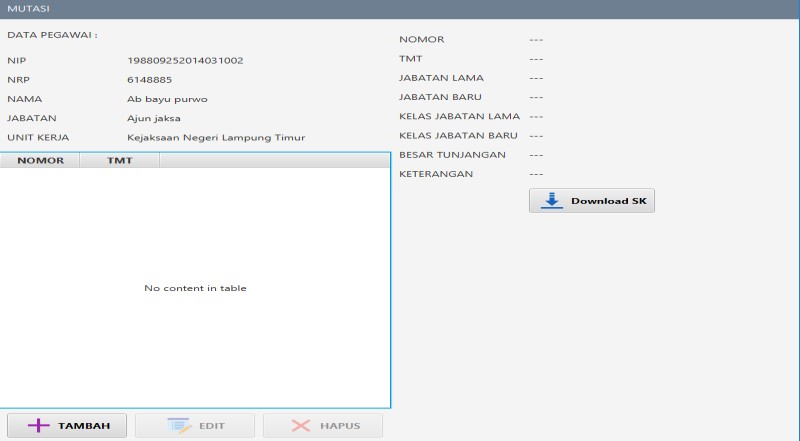
1. **Implementasi Input data mutasi**

Implementasi input data mutasi akan menambah data mutasi, maka sistem menampilkan halaman input data mutasi seperti pada gambar



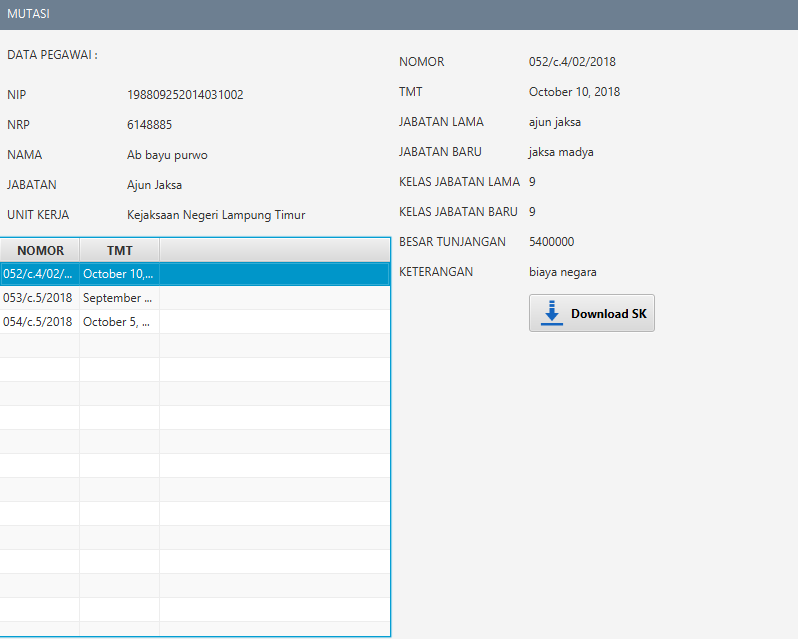
1. **Implementasi Proses Halaman Mutasi**

Halaman ini dipergunakan oleh Kaur Kepegawaian untuk dapat mengolah (input, edit, hapus, download) data mutasi pegawai ke dalam sistem.



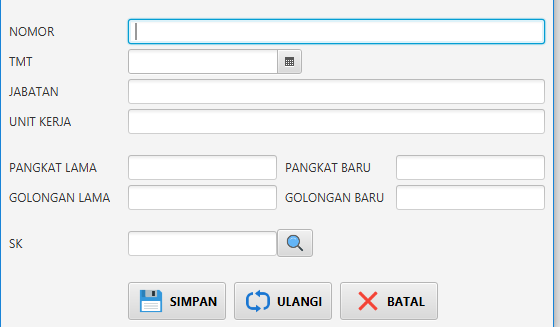
1. **`Implementasi output mutasi**

Implementasi ini hasil output dari data mutasi ke dalam sistem

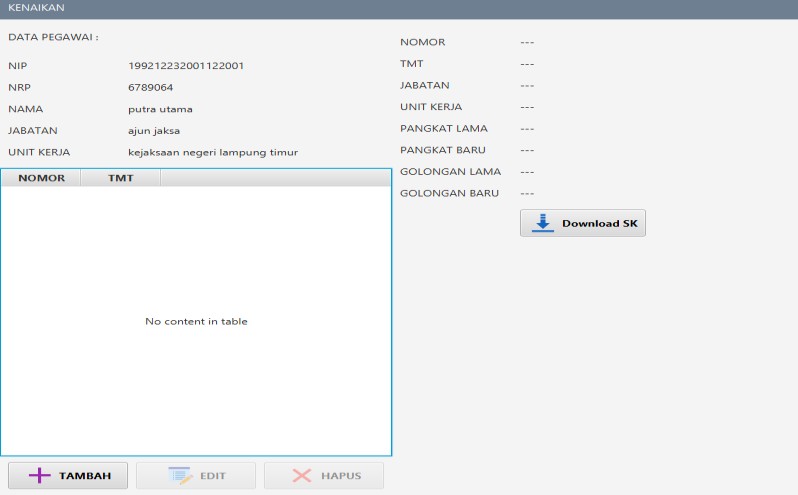


1. **Implementasi input data kenaikan pangkat**

Implementasi input data mutasi akan menambah data mutasi, maka sistem menampilkan halaman input data mutasi

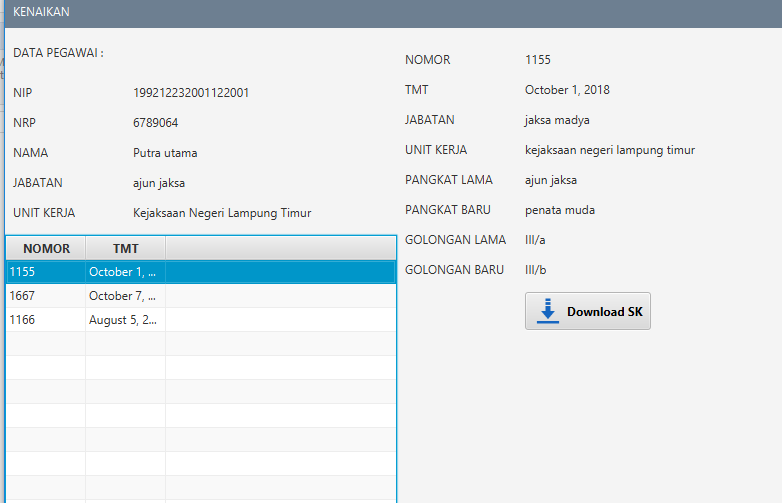


1. **Implementasi Proses Halaman Kenaikan Pangkat**

Halaman ini dipergunakan oleh Kaur Kepegawaian untuk dapat mengolah (input, edit, hapus, download) data kenaikan pangkat pegawai ke dalam sistem.

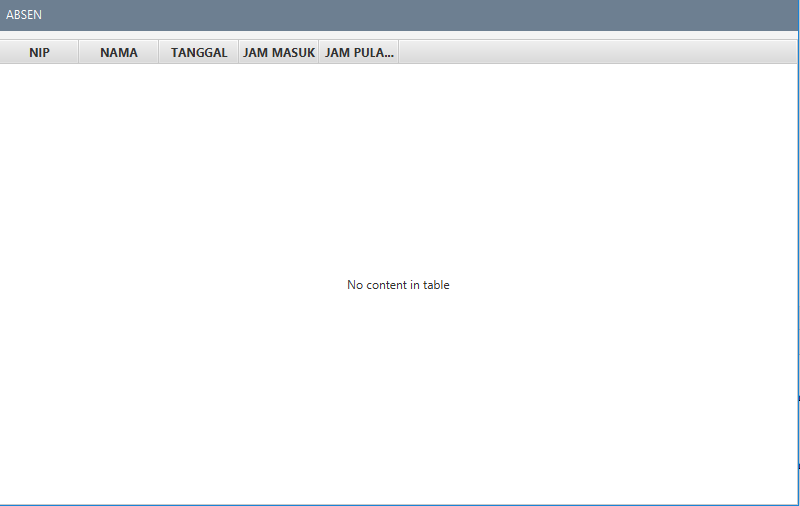
1. **Implementasi Output Data Kenaikan Pangkat**

Implementasi ini hasil output dari data kenaikan pangkat ke dalam sistem



1. **Implementasi Halaman Absen**

Halaman absen dibagi menjadi 2 (dua) bagian, yaitu halaman absen yang menampilkan halaman sebelum melakukan absensi pegawai seperti pada



**kesimpulan**

Adapun kesimpulan dari Sistem Informasi Kepegawaian pada Kejaksaan Negeri Lampung Timur adalah sebagai berikut :

1. Sistem yang dibangun dapat memudahkan presensi kehadiran Pegawai Kejaksaan dengan pemanfaatan smartphone Android yang didalamnya terdapat QR Code presensi.
2. Dapat memudahkan Kaur Kepegawaian dalam melakukan pengolahan data mutasi dan kenaikan pangkat pegawai termasuk mengolah data laporannya.

**package** dtc;

**import** mdl.AbsenMdl; **import** org.json.JSONArray; **import** org.json.JSONObject;

**import** java.util.ArrayList;

**import** java.util.List;

**public class** AbsenDtc { **private** String **partUrl**; **private** String **method**;

**private** JSONObject **jSONObject**; **private** Dtc **dtc** = **new** Dtc();

**private void** setPartUrlMethodObject(String instruction, AbsenMdl mdl) {

**jSONObject** = **new** JSONObject();

**switch** (instruction) {

**case "add"**:

**partUrl** = **"absen"**; **method** = **"POST"**;

**jSONObject**.put(**"tgl"**, mdl.**tgl**);

**if** (mdl.**jammasuk** != **null** && !mdl.**jammasuk**.equals(**""**))

**jSONObject**.put(**"jammasuk"**, mdl.**jammasuk**);

**if** (mdl.**jampulang** != **null** && !mdl.**jampulang**.equals(**""**))

**jSONObject**.put(**"jampulang"**, mdl.**jampulang**);

**jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**);

**break**; **case "upd"**:

**partUrl** = **"absen/"** + mdl.**idabsen**;

**method** = **"PUT"**;

**jSONObject**.put(**"tgl"**, mdl.**tgl**);

**if** (mdl.**jammasuk** != **null** && !mdl.**jammasuk**.equals(**""**))

**jSONObject**.put(**"jammasuk"**, mdl.**jammasuk**);

**if** (mdl.**jampulang** != **null** && !mdl.**jampulang**.equals(**""**))

**jSONObject**.put(**"jampulang"**, mdl.**jampulang**);

**jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**);

**break**; **case "del"**:

**partUrl** = **"absen/"** + mdl.**idabsen**;

**method** = **"DELETE"**; **break**;

**case "get"**:

**if** (mdl == **null**) { **partUrl** = **"absen"**; **method** = **"GET"**;

} **else** {

**partUrl** = **"absen/some"**; **method** = **"POST"**;

**jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**);

}

**break**;

}

}

**private** AbsenMdl setMdl(JSONObject jo) { AbsenMdl mdl = **new** AbsenMdl();

mdl.**idabsen** = Integer.*toString*(jo.getInt(**"idabsen"**));

mdl.**tgl** = jo.getString(**"tgl"**);

**if** (!jo.isNull(**"jammasuk"**)) mdl.**jammasuk** = jo.getString(**"jammasuk"**); **if** (!jo.isNull(**"jampulang"**)) mdl.**jampulang** = jo.getString(**"jampulang"**); mdl.**pegawai\_nip** = jo.getString(**"pegawai\_nip"**);

mdl.**nrp** = jo.getString(**"nrp"**); mdl.**nama** = jo.getString(**"nama"**);

**return** mdl;

}

**public int** aud(String instruction, AbsenMdl mdl) { setPartUrlMethodObject(instruction, mdl);

**return dtc**.aud(**partUrl**, **method**, **jSONObject**);

}

**public** List<AbsenMdl> get(AbsenMdl mdl) { List<AbsenMdl> mdls = **new** ArrayList<>();

setPartUrlMethodObject(**"get"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null**) {

JSONArray jsonArray = **new** JSONArray(inputLine);

**for** (**int** i = 0; i < jsonArray.length(); i++) { JSONObject jo = jsonArray.getJSONObject(i);

mdls.add(setMdl(jo));

}

}

**return** mdls;

}

}

**package** dtc;

**import** mdl.AdminMdl; **import** org.json.JSONArray; **import** org.json.JSONObject;

**import** java.util.ArrayList;

**import** java.util.List;

**public class** AdminDtc { **private** String **partUrl**; **private** String **method**;

**private** JSONObject **jSONObject**; **private** Dtc **dtc** = **new** Dtc();

**private void** setPartUrlMethodObject(String instruction, AdminMdl mdl) {

**jSONObject** = **new** JSONObject();

**switch** (instruction) {

**case "add"**:

**partUrl** = **"admin"**; **method** = **"POST"**;

**jSONObject**.put(**"username"**, mdl.**username**); **jSONObject**.put(**"password"**, mdl.**password**); **jSONObject**.put(**"nama"**, mdl.**nama**);

**break**; **case "upd"**:

**partUrl** = **"admin/"** + mdl.**idadmin**;

**method** = **"PUT"**;

**jSONObject**.put(**"username"**, mdl.**username**); **jSONObject**.put(**"password"**, mdl.**password**); **jSONObject**.put(**"nama"**, mdl.**nama**);

**break**; **case "del"**:

**partUrl** = **"admin/"** + mdl.**idadmin**;

**method** = **"DELETE"**; **break**;

**case "get"**:

**if** (mdl == **null**) { **partUrl** = **"admin"**; **method** = **"GET"**;

} **else** {

**partUrl** = **"admin/some"**; **method** = **"POST"**;

**jSONObject**.put(**"idadmin"**, mdl.**idadmin**);

}

**break**; **case "login"**:

**partUrl** = **"admin/login"**; **method** = **"POST"**;

**jSONObject**.put(**"username"**, mdl.**username**); **jSONObject**.put(**"password"**, mdl.**password**); **break**;

}

}

**private** AdminMdl setMdl(JSONObject jo) { AdminMdl mdl = **new** AdminMdl();

mdl.**idadmin** = Integer.*toString*(jo.getInt(**"idadmin"**)); mdl.**username** = jo.getString(**"username"**); mdl.**password** = jo.getString(**"password"**);

mdl.**nama** = jo.getString(**"nama"**);

**return** mdl;

}

**public int** aud(String instruction, AdminMdl mdl) { setPartUrlMethodObject(instruction, mdl);

**return dtc**.aud(**partUrl**, **method**, **jSONObject**);

}

**public** List<AdminMdl> get(AdminMdl mdl) { List<AdminMdl> mdls = **new** ArrayList<>();

setPartUrlMethodObject(**"get"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null**) {

JSONArray jsonArray = **new** JSONArray(inputLine);

**for** (**int** i = 0; i < jsonArray.length(); i++) { JSONObject jo = jsonArray.getJSONObject(i);

mdls.add(setMdl(jo));

}

}

**return** mdls;

}

**public** AdminMdl login(AdminMdl mdl) { AdminMdl m = **null**;

setPartUrlMethodObject(**"login"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null**) {

JSONObject jo = **new** JSONObject(inputLine);

m = setMdl(jo);

}

**return** m;

}

}

**package** dtc;

**import** mdl.ChartMdl; **import** org.json.JSONArray; **import** org.json.JSONObject;

**import** java.util.ArrayList;

**import** java.util.List;

**public class** ChartDtc { **private** String **partUrl**; **private** String **method**;

**private** JSONObject **jSONObject**; **private** Dtc **dtc** = **new** Dtc();

**private void** setPartUrlMethodObject(String instruction, ChartMdl mdl) {

**jSONObject** = **new** JSONObject();

**switch** (instruction) {

**case "chartpegawai"**:

**partUrl** = **"absen/chartpegawai"**; **method** = **"POST"**;

**jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**);

**break**;

}

}

**public** List<ChartMdl> getChartPegawai(ChartMdl mdl) { List<ChartMdl> mdls = **new** ArrayList<>();

setPartUrlMethodObject(**"chartpegawai"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null**) {

JSONArray jsonArray = **new** JSONArray(inputLine);

**for** (**int** i = 0; i < jsonArray.length(); i++) { JSONObject jo = jsonArray.getJSONObject(i);

ChartMdl m = **new** ChartMdl();

m.**month** = Integer.*toString*(jo.getInt(**"month"**)); m.**year** = Integer.*toString*(jo.getInt(**"year"**)); m.**count** = Integer.*toString*(jo.getInt(**"count"**)); m.**pegawai\_nip** = jo.getString(**"pegawai\_nip"**);

mdls.add(m);

}

}

**return** mdls;

}

}

**package** dtc;

**public class** CVDtc {

**private** AdminDtc **adminDtc** = **new** AdminDtc();

}

**package** dtc;

**import** cnn.Cnn;

**import** com.enwif.hcrr.HCRR; **import** hlp.InfoDialog; **import** org.json.JSONObject;

**import** java.awt.\*;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

**import** java.net.URI;

**import** java.net.http.HttpClient; **import** java.net.http.HttpRequest; **import** java.net.http.HttpResponse; **import** java.nio.file.Paths;

**class** Dtc {

**private** String audget(String partUrl, String method, JSONObject jSONObject) { HttpResponse<String> httpResponse = HCRR.*sendSync*(Cnn.***url*** + partUrl, method,

jSONObject.toString());

String r = httpResponse.body(); System.***out***.println(**"Dtc : "** + r);

**return** r;

}

**private void** adFile(String partUrl, String method, JSONObject jSONObject, String pathFile, String nameFile) {

HCRR.*sendSyncFile*(Cnn.***url*** + partUrl, method, jSONObject.toString(), pathFile, nameFile);

}

**int** aud(String partUrl, String method, JSONObject jSONObject) { String inputLine = audget(partUrl, method, jSONObject);

**if** (!inputLine.isEmpty()) {

**if** (inputLine.equals(**"invalid"**) || inputLine.equals(**"err"**)) { InfoDialog.*showDlg*(**"Error"**, inputLine);

**return** 0;

}

**else return** Integer.*parseInt*(inputLine);

} **else return** 0;

}

String get(String partUrl, String method, JSONObject jSONObject) { String inputLine = audget(partUrl, method, jSONObject);

**if** (!inputLine.isEmpty()) {

**if** (inputLine.equals(**"invalid"**) || inputLine.equals(**"err"**)) {

InfoDialog.*showDlg*(**"Error"**, inputLine);

**return null**;

}

**else return** inputLine;

} **else return null**;

}

**void** getFile(String partUrl, String method, JSONObject jSONObject, String pathFile, String nameFile) {

adFile(partUrl, method, jSONObject, pathFile, nameFile);

**try** {

Desktop.*getDesktop*().open(**new** File(pathFile + nameFile));

} **catch** (IOException e) { e.printStackTrace();

}

}

**public void** test(String partUrl, String method, JSONObject jSONObject) { String path = System.*getProperty*(**"user.home"**) + **"/Downloads/"**; String file = **"xxx.pdf"**;

HttpClient httpClient = HttpClient.*newHttpClient*(); HttpRequest httpRequest = **null**;

**try** {

httpRequest = HttpRequest.*newBuilder*()

.uri(URI.*create*(Cnn.***url*** + partUrl))

.header(**"Content-Type"**, **"multipart/form-data"**)

.method(method, HttpRequest.BodyPublishers.*ofFile*(Paths.*get*(path +

file)))

.build();

} **catch** (FileNotFoundException e) { e.printStackTrace();

}

**try** {

HttpResponse<String> httpResponse = httpClient.send(httpRequest, HttpResponse.BodyHandlers.*ofString*());

String r = httpResponse.body(); System.***out***.println(**"FileFileFile : "** + r);

} **catch** (IOException | InterruptedException e) { e.printStackTrace();

}

}

}

**package** dtc;

**import** mdl.FileMdl;

**import** org.json.JSONObject;

**import** java.time.LocalDate;

**public class** FileDtc { **private** String **partUrl**; **private** String **method**;

**private** JSONObject **jSONObject**; **private** String **pathFile**; **private** String **nameFile**;

**private** Dtc **dtc** = **new** Dtc();

**private void** setPartUrlMethodObject(String instruction, FileMdl mdl) {

**jSONObject** = **new** JSONObject();

**switch** (instruction) {

**case "addkenaikan"**:

**partUrl** = **"file/kenaikan"**; **method** = **"POST"**;

**jSONObject**.put(**"id"**, mdl.**id**); **jSONObject**.put(**"file"**, mdl.**fileByte64**); **break**;

**case "delkenaikan"**:

**partUrl** = **"file/kenaikan/test/"** + mdl.**id**;

**method** = **"DELETE"**; **break**;

**case "getkenaikan"**:

**partUrl** = **"file/kenaikan/"** + mdl.**id**;

**method** = **"GET"**;

**pathFile** = System.*getProperty*(**"user.home"**) + **"/Downloads/"**; **nameFile** = **"Kenaikan\_"** + LocalDate.*now*() + **".pdf"**;

**break**;

**case "addmutasi"**:

**partUrl** = **"file/mutasi"**; **method** = **"POST"**;

**jSONObject**.put(**"id"**, mdl.**id**); **jSONObject**.put(**"file"**, mdl.**fileByte64**); **break**;

**case "getmutasi"**:

**partUrl** = **"file/mutasi/"** + mdl.**id**;

**method** = **"GET"**;

**pathFile** = System.*getProperty*(**"user.home"**) + **"/Downloads/"**; **nameFile** = **"Mutasi\_"** + LocalDate.*now*() + **".pdf"**;

**break**;

}

}

**public int** aud(String instruction, FileMdl mdl) { setPartUrlMethodObject(instruction, mdl);

**return dtc**.aud(**partUrl**, **method**, **jSONObject**);

}

**public void** get(String instruction, FileMdl mdl) { setPartUrlMethodObject(instruction, mdl);

**dtc**.getFile(**partUrl**, **method**, **jSONObject**, **pathFile**, **nameFile**);

}

}

**package** dtc;

**import** mdl.KenaikanMdl; **import** org.json.JSONArray; **import** org.json.JSONObject;

**import** java.util.ArrayList;

**import** java.util.List;

**public class** KenaikanDtc { **private** String **partUrl**; **private** String **method**;

**private** JSONObject **jSONObject**; **private** Dtc **dtc** = **new** Dtc();

**private void** setPartUrlMethodObject(String instruction, KenaikanMdl mdl) {

**jSONObject** = **new** JSONObject();

**switch** (instruction) {

**case "add"**:

**partUrl** = **"kenaikan"**; **method** = **"POST"**;

**jSONObject**.put(**"nomor"**, mdl.**nomor**); **jSONObject**.put(**"tmt"**, mdl.**tmt**); **jSONObject**.put(**"jabatan"**, mdl.**jabatan**); **jSONObject**.put(**"unitkerja"**, mdl.**unitkerja**); **jSONObject**.put(**"pangkatlama"**, mdl.**pangkatlama**); **jSONObject**.put(**"pangkatbaru"**, mdl.**pangkatbaru**); **jSONObject**.put(**"golonganlama"**, mdl.**golonganlama**); **jSONObject**.put(**"golonganbaru"**, mdl.**golonganbaru**); **jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**); **break**;

**case "upd"**:

**partUrl** = **"kenaikan/"** + mdl.**idkenaikan**;

**method** = **"PUT"**;

**jSONObject**.put(**"nomor"**, mdl.**nomor**); **jSONObject**.put(**"tmt"**, mdl.**tmt**); **jSONObject**.put(**"jabatan"**, mdl.**jabatan**); **jSONObject**.put(**"unitkerja"**, mdl.**unitkerja**); **jSONObject**.put(**"pangkatlama"**, mdl.**pangkatlama**); **jSONObject**.put(**"pangkatbaru"**, mdl.**pangkatbaru**); **jSONObject**.put(**"golonganlama"**, mdl.**golonganlama**); **jSONObject**.put(**"golonganbaru"**, mdl.**golonganbaru**); **jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**); **break**;

**case "del"**:

**partUrl** = **"kenaikan/"** + mdl.**idkenaikan**;

**method** = **"DELETE"**; **break**;

**case "get"**:

**if** (mdl == **null**) { **partUrl** = **"kenaikan"**; **method** = **"GET"**;

} **else** {

**partUrl** = **"kenaikan/some"**; **method** = **"POST"**;

**jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**);

}

**break**;

}

}

**private** KenaikanMdl setMdl(JSONObject jo) { KenaikanMdl mdl = **new** KenaikanMdl();

mdl.**idkenaikan** = Integer.*toString*(jo.getInt(**"idkenaikan"**)); mdl.**nomor** = jo.getString(**"nomor"**);

mdl.**tmt** = jo.getString(**"tmt"**); mdl.**jabatan** = jo.getString(**"jabatan"**);

mdl.**unitkerja** = jo.getString(**"unitkerja"**); mdl.**pangkatlama** = jo.getString(**"pangkatlama"**); mdl.**pangkatbaru** = jo.getString(**"pangkatbaru"**); mdl.**golonganlama** = jo.getString(**"golonganlama"**); mdl.**golonganbaru** = jo.getString(**"golonganbaru"**); mdl.**pegawai\_nip** = jo.getString(**"pegawai\_nip"**);

**return** mdl;

}

**public int** aud(String instruction, KenaikanMdl mdl) { setPartUrlMethodObject(instruction, mdl);

**return dtc**.aud(**partUrl**, **method**, **jSONObject**);

}

**public** List<KenaikanMdl> get(KenaikanMdl mdl) { List<KenaikanMdl> mdls = **new** ArrayList<>();

setPartUrlMethodObject(**"get"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null**) {

JSONArray jsonArray = **new** JSONArray(inputLine);

**for** (**int** i = 0; i < jsonArray.length(); i++) { JSONObject jo = jsonArray.getJSONObject(i);

mdls.add(setMdl(jo));

}

}

**return** mdls;

}

}

**package** dtc;

**import** mdl.MutasiMdl; **import** org.json.JSONArray; **import** org.json.JSONObject;

**import** java.util.ArrayList;

**import** java.util.List;

**public class** MutasiDtc { **private** String **partUrl**; **private** String **method**;

**private** JSONObject **jSONObject**; **private** Dtc **dtc** = **new** Dtc();

**private void** setPartUrlMethodObject(String instruction, MutasiMdl mdl) {

**jSONObject** = **new** JSONObject();

**switch** (instruction) {

**case "add"**:

**partUrl** = **"mutasi"**; **method** = **"POST"**;

**jSONObject**.put(**"nomor"**, mdl.**nomor**); **jSONObject**.put(**"tmt"**, mdl.**tmt**); **jSONObject**.put(**"jabatanlama"**, mdl.**jabatanlama**);

**jSONObject**.put(**"jabatanbaru"**, mdl.**jabatanbaru**); **jSONObject**.put(**"kelasjabatanlama"**, mdl.**kelasjabatanlama**); **jSONObject**.put(**"kelasjabatanbaru"**, mdl.**kelasjabatanbaru**);

**if** (mdl.**besartunjangan** != **null** && !mdl.**besartunjangan**.equals(**""**))

**jSONObject**.put(**"besartunjangan"**, mdl.**besartunjangan**);

**if** (mdl.**keterangan** != **null** && !mdl.**keterangan**.equals(**""**))

**jSONObject**.put(**"keterangan"**, mdl.**keterangan**);

**jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**);

**break**; **case "upd"**:

**partUrl** = **"mutasi/"** + mdl.**idmutasi**;

**method** = **"PUT"**;

**jSONObject**.put(**"nomor"**, mdl.**nomor**); **jSONObject**.put(**"tmt"**, mdl.**tmt**); **jSONObject**.put(**"jabatanlama"**, mdl.**jabatanlama**); **jSONObject**.put(**"jabatanbaru"**, mdl.**jabatanbaru**);

**jSONObject**.put(**"kelasjabatanlama"**, mdl.**kelasjabatanlama**);

**jSONObject**.put(**"kelasjabatanbaru"**, mdl.**kelasjabatanbaru**);

**if** (mdl.**besartunjangan** != **null** && !mdl.**besartunjangan**.equals(**""**))

**jSONObject**.put(**"besartunjangan"**, mdl.**besartunjangan**);

**if** (mdl.**keterangan** != **null** && !mdl.**keterangan**.equals(**""**))

**jSONObject**.put(**"keterangan"**, mdl.**keterangan**);

**jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**);

**break**; **case "del"**:

**partUrl** = **"mutasi/"** + mdl.**idmutasi**;

**method** = **"DELETE"**; **break**;

**case "get"**:

**if** (mdl == **null**) { **partUrl** = **"mutasi"**; **method** = **"GET"**;

} **else** {

**partUrl** = **"mutasi/some"**; **method** = **"POST"**;

**jSONObject**.put(**"pegawai\_nip"**, mdl.**pegawai\_nip**);

}

**break**;

}

}

**private** MutasiMdl setMdl(JSONObject jo) { MutasiMdl mdl = **new** MutasiMdl();

mdl.**idmutasi** = Integer.*toString*(jo.getInt(**"idmutasi"**)); mdl.**nomor** = jo.getString(**"nomor"**);

mdl.**tmt** = jo.getString(**"tmt"**);

mdl.**jabatanlama** = jo.getString(**"jabatanlama"**); mdl.**jabatanbaru** = jo.getString(**"jabatanbaru"**); mdl.**kelasjabatanlama** = jo.getString(**"kelasjabatanlama"**); mdl.**kelasjabatanbaru** = jo.getString(**"kelasjabatanbaru"**); **if** (!jo.isNull(**"besartunjangan"**)) mdl.**besartunjangan** =

Integer.*toString*(jo.getInt(**"besartunjangan"**));

**if** (!jo.isNull(**"keterangan"**)) mdl.**keterangan** = jo.getString(**"keterangan"**); mdl.**pegawai\_nip** = jo.getString(**"pegawai\_nip"**);

**return** mdl;

}

**public int** aud(String instruction, MutasiMdl mdl) { setPartUrlMethodObject(instruction, mdl);

**return dtc**.aud(**partUrl**, **method**, **jSONObject**);

}

**public** List<MutasiMdl> get(MutasiMdl mdl) { List<MutasiMdl> mdls = **new** ArrayList<>();

setPartUrlMethodObject(**"get"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null**) {

JSONArray jsonArray = **new** JSONArray(inputLine);

**for** (**int** i = 0; i < jsonArray.length(); i++) { JSONObject jo = jsonArray.getJSONObject(i);

mdls.add(setMdl(jo));

}

}

**return** mdls;

}

}

**package** dtc;

**import** mdl.PegawaiMdl; **import** org.json.JSONArray; **import** org.json.JSONObject;

**import** java.util.ArrayList;

**import** java.util.List;

**public class** PegawaiDtc { **private** String **partUrl**; **private** String **method**;

**private** JSONObject **jSONObject**; **private** Dtc **dtc** = **new** Dtc();

**private void** setPartUrlMethodObject(String instruction, PegawaiMdl mdl) {

**jSONObject** = **new** JSONObject();

**switch** (instruction) {

**case "add"**:

**partUrl** = **"pegawai"**; **method** = **"POST"**;

**jSONObject**.put(**"nip"**, mdl.**nip**); **jSONObject**.put(**"nrp"**, mdl.**nrp**); **jSONObject**.put(**"nama"**, mdl.**nama**); **jSONObject**.put(**"tmplhr"**, mdl.**tmplhr**); **jSONObject**.put(**"tgllhr"**, mdl.**tgllhr**); **jSONObject**.put(**"jabatan"**, mdl.**jabatan**); **jSONObject**.put(**"unitkerja"**, mdl.**unitkerja**); **jSONObject**.put(**"golongan"**, mdl.**golongan**); **jSONObject**.put(**"email"**, mdl.**email**); **jSONObject**.put(**"nohp"**, mdl.**nohp**); **jSONObject**.put(**"status"**, mdl.**status**); **break**;

**case "upd"**:

**partUrl** = **"pegawai/"** + mdl.**old\_nip**;

**method** = **"PUT"**;

**jSONObject**.put(**"nip"**, mdl.**nip**);

**jSONObject**.put(**"nrp"**, mdl.**nrp**);

**jSONObject**.put(**"nama"**, mdl.**nama**); **jSONObject**.put(**"tmplhr"**, mdl.**tmplhr**); **jSONObject**.put(**"tgllhr"**, mdl.**tgllhr**); **jSONObject**.put(**"jabatan"**, mdl.**jabatan**); **jSONObject**.put(**"unitkerja"**, mdl.**unitkerja**); **jSONObject**.put(**"golongan"**, mdl.**golongan**); **jSONObject**.put(**"email"**, mdl.**email**); **jSONObject**.put(**"nohp"**, mdl.**nohp**); **jSONObject**.put(**"status"**, mdl.**status**); **break**;

**case "del"**:

**partUrl** = **"pegawai/"** + mdl.**nip**;

**method** = **"DELETE"**; **break**;

**case "get"**:

**if** (mdl == **null**) { **partUrl** = **"pegawai"**; **method** = **"GET"**;

} **else** {

**partUrl** = **"pegawai/some"**; **method** = **"POST"**;

**jSONObject**.put(**"nip"**, mdl.**nip**);

}

**break**; **case "src"**:

**partUrl** = **"pegawai/src"**; **method** = **"POST"**;

**jSONObject**.put(**"nip"**, mdl.**nip**); **jSONObject**.put(**"nama"**, mdl.**nama**); **break**;

}

}

**private** PegawaiMdl setMdl(JSONObject jo) { PegawaiMdl mdl = **new** PegawaiMdl(); mdl.**nip** = jo.getString(**"nip"**);

mdl.**nrp** = jo.getString(**"nrp"**); mdl.**nama** = jo.getString(**"nama"**); mdl.**tmplhr** = jo.getString(**"tmplhr"**); mdl.**tgllhr** = jo.getString(**"tgllhr"**); mdl.**jabatan** = jo.getString(**"jabatan"**);

mdl.**unitkerja** = jo.getString(**"unitkerja"**); mdl.**golongan** = jo.getString(**"golongan"**); mdl.**email** = jo.getString(**"email"**); mdl.**nohp** = jo.getString(**"nohp"**); mdl.**status** = jo.getString(**"status"**);

**return** mdl;

}

**public int** aud(String instruction, PegawaiMdl mdl) { setPartUrlMethodObject(instruction, mdl);

**return dtc**.aud(**partUrl**, **method**, **jSONObject**);

}

**public** List<PegawaiMdl> get(PegawaiMdl mdl) { List<PegawaiMdl> mdls = **new** ArrayList<>();

setPartUrlMethodObject(**"get"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null**) {

JSONArray jsonArray = **new** JSONArray(inputLine);

**for** (**int** i = 0; i < jsonArray.length(); i++) { JSONObject jo = jsonArray.getJSONObject(i);

mdls.add(setMdl(jo));

}

}

**return** mdls;

}

**public** List<PegawaiMdl> src(PegawaiMdl mdl) { List<PegawaiMdl> mdls = **new** ArrayList<>();

setPartUrlMethodObject(**"src"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null**) {

JSONArray jsonArray = **new** JSONArray(inputLine);

**for** (**int** i = 0; i < jsonArray.length(); i++) { JSONObject jo = jsonArray.getJSONObject(i);

mdls.add(setMdl(jo));

}

}

**return** mdls;

}

}

**package** dtc;

**import** mdl.QRAbsenMdl;

**import** org.json.JSONObject;

**public class** QRAbsenDtc { **private** String **partUrl**; **private** String **method**;

**private** JSONObject **jSONObject**; **private** Dtc **dtc** = **new** Dtc();

**private void** setPartUrlMethodObject(String instruction, QRAbsenMdl mdl) {

**jSONObject** = **new** JSONObject();

**switch** (instruction) {

**case "init"**:

**partUrl** = **"qrabsen/init"**; **method** = **"POST"**;

**break**; **case "qr"**:

**partUrl** = **"qrabsen/qr"**; **method** = **"POST"**;

**jSONObject**.put(**"qr"**, mdl.**qr**);

**break**; **case "get"**:

**partUrl** = **"qrabsen/getverify"**; **method** = **"POST"**;

**break**;

}

}

**private** QRAbsenMdl setMdl(JSONObject jo) { QRAbsenMdl mdl = **new** QRAbsenMdl(); mdl.**nip** = jo.getString(**"nip"**);

mdl.**nrp** = jo.getString(**"nrp"**); mdl.**nama** = jo.getString(**"nama"**); mdl.**jabatan** = jo.getString(**"jabatan"**);

mdl.**unitkerja** = jo.getString(**"unitkerja"**);

**return** mdl;

}

**public int** aud(String instruction, QRAbsenMdl mdl) { setPartUrlMethodObject(instruction, mdl);

**return dtc**.aud(**partUrl**, **method**, **jSONObject**);

}

**public** QRAbsenMdl get() { QRAbsenMdl mdl = **null**;

setPartUrlMethodObject(**"get"**, mdl);

String inputLine = **dtc**.get(**partUrl**, **method**, **jSONObject**); **if** (inputLine != **null** && !inputLine.equals(**"0"**)) {

JSONObject jo = **new** JSONObject(inputLine);

mdl = setMdl(jo);

}

**return** mdl;

}

}

**package** mdl;

**import** java.time.Instant;

**import** java.time.ZoneId;

**import** java.time.format.DateTimeFormatter;

**import** java.time.format.FormatStyle;

**public class** AbsenMdl {

**public** String **idabsen**, **tgl**, **jammasuk**, **jampulang**, **pegawai\_nip**; **public** String **nrp**, **nama**;

**public** String getIdabsen() {

**return idabsen**;

}

**public** String getTgl() {

**return tgl**;

}

**public** String getJammasuk() {

**return jammasuk**;

}

**public** String getJampulang() {

**return jampulang**;

}

**public** String getPegawai\_nip() {

**return pegawai\_nip**;

}

**public** String getNrp() {

**return nrp**;

}

**public** String getNama() {

**return nama**;

}

**public** String getVw\_tgl() {

**return** DateTimeFormatter.*ofLocalizedDate*(FormatStyle.***LONG***).format(Instant.*parse*(**tgl**).atZone(Z oneId.*systemDefault*()));

}

}

**package** mdl;

**import** java.time.Instant;

**import** java.time.ZoneId;

**import** java.time.format.DateTimeFormatter;

**import** java.time.format.FormatStyle;

**public class** KenaikanMdl {

**public** String **idkenaikan**, **nomor**, **tmt**, **jabatan**, **unitkerja**, **pangkatlama**, **pangkatbaru**, **golonganlama**, **golonganbaru**, **pegawai\_nip**;

**public** String getIdkenaikan() {

**return idkenaikan**;

}

**public** String getNomor() {

**return nomor**;

}

**public** String getTmt() {

**return tmt**;

}

**public** String getJabatan() {

**return jabatan**;

}

**public** String getUnitkerja() {

**return unitkerja**;

}

**public** String getPangkatlama() {

**return pangkatlama**;

}

**public** String getPangkatbaru() {

**return pangkatbaru**;

}

**public** String getGolonganlama() {

**return golonganlama**;

}

**public** String getGolonganbaru() {

**return golonganbaru**;

}

**public** String getPegawai\_nip() {

**return pegawai\_nip**;

}

**public** String getVw\_tmt() {

**return** DateTimeFormatter.*ofLocalizedDate*(FormatStyle.***LONG***).format(Instant.*parse*(**tmt**).atZone(Z oneId.*systemDefault*()));

}

}

**package** mdl;

**import** java.time.Instant;

**import** java.time.ZoneId;

**import** java.time.format.DateTimeFormatter;

**import** java.time.format.FormatStyle;

**public class** MutasiMdl {

**public** String **idmutasi**, **nomor**, **tmt**, **jabatanlama**, **jabatanbaru**, **kelasjabatanlama**, **kelasjabatanbaru**, **besartunjangan**, **keterangan**, **pegawai\_nip**;

**public** String getIdmutasi() {

**return idmutasi**;

}

**public** String getNomor() {

**return nomor**;

}

**public** String getTmt() {

**return tmt**;

}

**public** String getJabatanlama() {

**return jabatanlama**;

}

**public** String getJabatanbaru() {

**return jabatanbaru**;

}

**public** String getKelasjabatanlama() {

**return kelasjabatanlama**;

}

**public** String getKelasjabatanbaru() {

**return kelasjabatanbaru**;

}

**public** String getBesartunjangan() {

**return besartunjangan**;

}

**public** String getKeterangan() {

**return keterangan**;

}

**public** String getPegawai\_nip() {

**return pegawai\_nip**;

}

**public** String getVw\_tmt() {

**return** DateTimeFormatter.*ofLocalizedDate*(FormatStyle.***LONG***).format(Instant.*parse*(**tmt**).atZone(Z oneId.*systemDefault*()));

}

}

**package** mdl;

**import** java.time.Instant;

**import** java.time.ZoneId;

**import** java.time.format.DateTimeFormatter;

**import** java.time.format.FormatStyle;

**public class** PegawaiMdl {

**public** String **nip**, **nrp**, **nama**, **tmplhr**, **tgllhr**, **jabatan**, **unitkerja**, **golongan**, **email**, **nohp**, **status**;

**public** String **old\_nip**;

**public** String getNip() {

**return nip**;

}

**public** String getNrp() {

**return nrp**;

}

**public** String getNama() {

**return nama**;

}

**public** String getTmplhr() {

**return tmplhr**;

}

**public** String getTgllhr() {

**return tgllhr**;

}

**public** String getJabatan() {

**return jabatan**;

}

**public** String getUnitkerja() {

**return unitkerja**;

}

**public** String getGolongan() {

**return golongan**;

}

**public** String getEmail() {

**return email**;

}

**public** String getNohp() {

**return nohp**;

}

**public** String getStatus() {

**return status**;

}

**public** String getOld\_nip() {

**return old\_nip**;

}

**public** String getVw\_tgllhr() {

**return** DateTimeFormatter.*ofLocalizedDate*(FormatStyle.***LONG***).format(Instant.*parse*(**tgllhr**).atZon e(ZoneId.*systemDefault*()));

}

**public** String getVw\_status() {

**switch** (**status**) {

**case "1"**:

**return "Aktif"**; **case "0"**:

**return "Tidak Aktif"**; **default**:

**return "Tidak Aktif"**;

}

}

}

**package** mdl;

**public class** QRAbsenMdl {

**public** String **qr**;

**public** String **nip**, **nrp**, **nama**, **jabatan**, **unitkerja**;