Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Team project ©2017
Dony Pratidana S. Hum | Bima Agus Setyawan S. IIP

BAB III
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pada saat pelaksanaan kegiatan kerja magang kedudukan yang diberikan adalah *IT Programmer* divisi *finance* (keuangan). PT Pharos Indonesia membagi *IT Programmer* menjadi beberapa divisi, ada divisi yang menangani gudang untuk mencatat stok-stok yang tersedia pada gudang, divisi pabrik untuk mencatat produk-produk yang dihasilkan, divisi *human resource* untuk mencatat data karyawan dan keuangan . *IT Programmer* pada divisi *finance* lebih fokus terhadap pembuatan aplikasi untuk membantu perhitungan dan pemrosesan data para akuntan yang bekerja di PT Pharos Indonesia.

Koordinasi yang dilakukan, divisi *accounting* biasanya akan melakukan permintaan kepada manager *IT Programmer* divisi keuangan untuk membuat sebuah aplikasi yang membantu mereka dalam melakukan pemrosesan data. *Manager IT Programmer* akan membagi tugas kepada para anggota yang ada pada divisi tersebut dan membantu apabila ada anggota yang mengalami kesulitan. *Manager* juga membuat permintaan untuk hak akses sistem basis data kepada divisi *Database Administrator* apabila ada tabel dalam sistem basis data yang tidak dapat diakses karena tidak mempunyai hak akses.

Selama mengerjakan tugasnya, *IT Programmer* juga selalu menghubungi divisi *Database Administrator* apabila terdapat permintaan untuk melakukan *rollback* pada sistem basis data yang diminta karena terdapat kesalahan yang dilakukan pada saat proses pembuatan aplikasi.

Ketika aplikasi sudah selesai dibuat, tahap selanjutnya adalah mengadakan pertemuan dengan *IT Manager* yang menangani semua divisi yang ada di PT Pharos Indonesia untuk mengadakan demo terhadap aplikasi yang sudah dibuat. Jika proses demo sudah selesai maka aplikasi dapat didistribusikan kepada user.

Jika suatu saat aplikasi yang dipakai user mengalami masalah. User akan menghubungi *Manager IT Programmer* dengan divisi yang bersangkutan, kemudian

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

Manager tersebut beserta anggota yang membuat aplikasi tersebut akan melakukan proses *debugging* untuk menyelesaikan masalah pada aplikasi itu.

3.2 Tugas yang Dilakukan

Selama pelaksanaan kerja magang yang berlangsung selama 41 hari, terdapat dua modul yang dikerjakan, yaitu modul Master Category yang digunakan karyawan untuk menambahkan kategori produk serta melihat produk sesuai dengan kategori yang ada dan Master Principal yang menangani informasi *principal* (perusahaan yang menjalin hubungan dengan PT Pharos Indonesia) mulai dari informasi mengenai identitas perusahaan sampai data faktur pajak dan keuangan. Master Principal juga memperlihatkan informasi secara lengkap daftar produk yang ditawarkan oleh masing-masing *principal*.

Aplikasi yang dikerjakan terlebih dahulu adalah Master Category. Setelah pengerjaan Master Category sudah selesai, aplikasi dipresentasikan pada Direktur PT Pharos Indonesia. Setelah Master Category dipresentasikan atau sudah diuji bahwa aplikasi tersebut tidak mengalami kendala maka PT Pharos Indonesia memberikan tugas untuk mengerjakan Master Principal.

Spesifikasi modul Master Category adalah sebagai berikut.

1. Menampilkan kategori produk yang sudah ada dilengkapi dengan tombol *next* dan *previous* untuk mengarahkan ke data selanjutnya atau sebelumnya hal ini diperlukan untuk menghindari penggunaan waktu yang lama untuk menampilkan data kategori yang jumlahnya cukup banyak.
2. Melakukan pencarian kategori berdasarkan nama atau kode dan ditampilkan dalam bentuk tabel yang dapat diurutkan sesuai kode atau nama. Item pada tabel dapat dipilih dan akan ditampilkan pada halaman awal Master Category.
3. Mencetak daftar kategori dan daftar produk pada kategori yang dipilih dari hasil pencarian.
4. Melakukan penambahan dan menoaktifkan data kategori dilengkapi dengan konfirmasi atau notifikasi untuk memastikan aksi yang dilakukan.

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

5. Setiap bentuk pengisian formulir dilengkapi dengan validasi masukan dari pengguna (*input*).
6. Pemeriksaan hak akses yang dilakukan sebelum pengguna melakukan *action* seperti menghapus, mengubah, dan menambah data.
7. Modul dibuat dengan tampilan responsif.

Spesifikasi modul Master Principal adalah sebagai berikut.

1. Menampilkan informasi lengkap *principal* dikelompokkan berdasarkan kategori (*finance*, faktur pajak, *product list*) dalam bentuk *tab panel*.
2. Tombol *next* dan *previous* untuk navigasi ke data *principal* selanjutnya dan sebelumnya.
3. Menambah, mengubah, mengaktifkan dan menonaktifkan data *principal*.
4. Melakukan pencarian berdasarkan keyword berupa nama atau kode *principal* yang dimasukkan. Hasil pencarian dalam bentuk tabel dan dapat diurutkan. Data pada tabel dapat diklik dan akan ditampilkan pada halaman utama Master Principal.
5. Mencetak daftar produk yang disediakan masing-masing *principal*.
6. Mengubah data *finance* setiap *principal*.
7. Mengubah data faktur pajak *principal*.
8. Melakukan verifikasi dan validasi hak akses user.
9. Modul dibuat sesuai seperti aplikasi *desktop* yang dibangun dengan Visual Basic 6.0.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

3.3 Uraian Pelaksanaan Kerja Magang

Berikut ini rincian pekerjaan yang dilakukan dilaksanakan di PT Pharos Indonesia.

Minggu Ke-	Pekerjaan yang Dilakukan
1	• Menganalisa dan melakukan perencanaan pembangunan *web based application* dan menentukan *software, hardware, framework* apa saja yang dibutuhkan. • Mempersiapkan peralatan dan sumber daya yang diperlukan untuk membangun *web based application* (CodeIgniter, Bootstrap, Datatable, dan AngularJS) • Melakukan konfigurasi agar aplikasi dapat terhubung dengan SQLServer 2000
2	• Melakukan desain tampilan Master Category • Membangun *controller* dan model Master Category dengan menggunakan *tools-tools* yang sudah dipersiapkan
3	• Melakukan pengujian modul Master Category pada *web* yang dibangun • Mempresentasikan aplikasi kepada Direktur PT Pharos Indonesia • Melakukan revisi dari hasil presentasi yang dilakukan
4	• Melakukan konfigurasi komputer *Windows 7* untuk dapat

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

	menjalankan aplikasi yang dibuat sebelumnya pada sistem operasi Windows XP • Mengubah konfigurasi CodeIgniter untuk dapat terhubung dengan SQL Server 2008
5	• Desain tampilan Master Principal • Mengimplementasi hak akses dan validasi masukan dari pengguna
6	• Membuat model Principal (CodeIgniter) • Membangun *controller* Master Principal (CodeIgniter) • Mengimplementasikan operasi dan fitur-fitur Master Principal
7	• Membuat *factory* pada *framework* AngularJS untuk dapat mengakses *controller* Principal yang sudah dibuat. • Mengimplementasikan fungsi menambah dan mengubah data *principal*. • Mengimplementasikan fungsi mencetak daftar list produk
8	• Membuat halaman data faktur pajak dan *finance* • Melakukan testing Master Principal • Mempresentasikan aplikasi yang sudah dibuat

Tabel 3.1 Pekerjaan Per Minggu

Pada minggu pertama dilakukan perencanaan dengan melakukan pembacaan pada kode program yang ditulis dengan bahasa Virtual Basic 6.0 dan menanyakan

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

kepada *programmer* lainnya apabila ada kode program yang tidak dimengerti. Setelah tahap perencanaan selesai, dilanjutkan dengan mencatat daftar software yang nantinya akan digunakan untuk membangun aplikasi. Setelah semua *software* sudah dipersiapkan hal yang dilakukan adalah meminta izin kepada PT Pharos Indonesia untuk memasang daftar perangkat lunak yang diperlukan ke komputer milik Pharos Indonesia.

Berikut adalah daftar peralatan yang dibutuhkan untuk membangun Master Category dan Master Principal

1. XAMPP versi 1.8.2 untuk Windows XP.
2. Bootstrap 3.3.6 sebagai CSS Framework untuk membuat tampilan *web* yang responsif.
3. Notepad++ 6.9.2 sebagai *text editor*.
4. AngularDataTables yang library yang merupakan sebuah *directive* pada AngularJS untuk menampilkan dataTables.
5. SQLSRV Driver versi 2.0, 3.0, 3.1, dan 3.2 untuk koneksi ke SQLServer.
6. CodeIgniter 3.2 sebagai PHP MVC Framework.
7. AngularJS 1.2 untuk melakukan *request* pada server secara asinkronus.

Sesudah perangkat lunak yang diperlukan tersedia maka dilakukan pengujian dengan membuat aplikasi sederharna sebelum membuat Master Category ke *server* Genesis yang merupakan *server database* SQLServer 2000. Koneksi dilakukan dari komputer dengan *operating system* Windows XP. PT Pharos Indonesia menggunakan sistem operasi Windows XP karena sebagian besar sistem PT Pharos Indonesia merupakan program yang dibangun dengan menggunakan Visual Basic 6.0.

Pada minggu kedua, dimulai dengan membuat desain tampilan *web* untuk Master Category kemudian dilanjutkan dengan membuat fungsi-fungsi pada *controller* Codeigniter seperti pencarian, pencetakan daftar produk, penambahan kategori dan penoaktifan data. Fungsi-fungsi untuk melakukan operasi *database* dibuat ke dalam satu model sehingga nantinya *controller* akan memanggil model yang telah dibuat untuk melakukan *query* dan setelah model memberikan hasil *query*,

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

controller bertugas untuk melakukan transfer hasil *query* ke halaman *view* (tampilan). Pada minggu ketiga, dilakukan pengujian pada Master Category dengan cara menjalankan semua fungsionalitas yang tersedia pada Master Category. Setelah pengujian telah selesai dipresentasikan pada Direktur PT Pharos Indonesia hal selanjutnya adalah melakukan revisi pada Master Category untuk meningkatkan *user experience* ketika menggunakan aplikasi tersebut.

Pada minggu keempat, PT Pharos Indonesia memutuskan untuk menggunakan sistem operasi Windows 7 dan SQL Server 2008 sehingga konfigurasi ulang dilakukan agar komputer bisa terkoneksi dengan Server yang baru. Konfigurasi ulang diawali dengan mengubah versi XAMPP untuk sistem operasi Windows 7 dan mengganti *driver* sql_srv yang digunakan untuk dapat terhubung ke SQL Server 2008.

Pada minggu kelima, Ibu Susan selaku manager IT untuk divisi finance memberikan tugas untuk membuat Master Principal menjadi aplikasi berbasis *web*. Master Principal merupakan sebuah program untuk menampilkan data-data principal. Pekerjaan dimulai dengan membuat tampilan utama dan formulir-formulir untuk menambahkan atau mengubah data principal dilengkapi dengan fitur validasi input dengan menggunakan AngularJS. Validasi hak akses user untuk menambah,mengubah data principal juga diimplementasikan pada aplikasi yang dibuat.

Pada minggu keenam, pekerjaan yang dilakukan adalah membuat model yang berisikan fungsi-fungsi untuk memanggil *stored procedure* yang telah dibuat sebelumnya oleh *programmer* PT Pharos Indonesia yang ada di dalam *database* SQL Server. Setelah model dibuat, pekerjaan selanjutnya adalah membuat *controller* yang berisi fungi-fungsi yang merupakan fungsionalitas dari Master Principal.

Pada minggu ketutuh, pekerjaan yang dilakukan adalah mengimplementasikan Factory pada *controller* AngularJS untuk memanggil fungsi-fungsi yang telah dibuat oleh *controller* CodeIgniter. *Controller* CodeIgniter menyediakan data dalam bentuk

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

JSON dari pemanggilan fungsi pada model yang telah dibuat. Request yang dilakukan oleh AngularJS terjadi secara asinkronus sehingga tampilan *web* tidak perlu digambar kembali. Di minggu ini juga dibuat halaman untuk menampilkan daftar produk dari masing-masing *principal* yang dapat dicetak dalam bentuk apapun dengan menggunakan library yang disediakan oleh *datatables.net*.

Pada minggu kedelapan, pekerjaan yang dilakukan adalah membuat halaman untuk menampilkan data *finance* dan faktur pajak masing-masing principal. Setelah selesai, dilakukan *testing* dengan menjalankan semua fungsionalitas yang tersedia pada Master Principal. Setelah selesai maka aplikasi dipresentasikan kembali. Setelah modul-modul ini diuji, direvisi, dan diperbaiki, aplikasi berbasis web yang telah dibuat akan di-*hosting* pada suatu komputer dan dapat diakses oleh *user* yang menggunakannya melalui alamat IP atau URL *web* komputer *host*-nya.

3.3.1 Proses Pelaksanaan

Pelaksanaan kerja magang menggunakan metode *System Development Life Cycle* (SDLC). Dari banyak model SDLC yang tersedia, model SDLC yang digunakan adalah *Agile Model* yang merupakan model yang terdiri dari beberapa iterasi yang dilakukan untuk peningkatan terhadap kepuasan *customer* berdasarkan *requirement* yang didefinisikan. *Agile Model* digunakan karena *requirement* awal didefinisikan secara jelas oleh *client* dan terjadi beberapa kali perubahan untuk meningkatkan performa sistem sehingga model ini tepat untuk diimplementasikan pada projek *web based application* ini. Pada akhir proses iterasi, produk dipresentasikan ke *client* atau *stakeholders*.

Pengerjaan Master Principal dan Master Category dibagi menjadi dua iterasi. Tahap iterasi pertama dilakukan dengan pembuatan aplikasi dengan menekankan pada fungsionalitas yang sesuai dengan *user requirement* yang telah diminta oleh PT Pharos Indonesia. Tahap Iterasi kedua ditujukan untuk meningkatkan *user experience* dengan cara membuat *request* yang dilakukan oleh aplikasi ke *database server* dapat berjalan secara asinkronus dengan menggunakan AngularJS sehingga aplikasi yang dibangun berbasis *web* tidak jauh berbeda dengan aplikasi *desktop* yang telah dibuat

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

sebelumnya dengan menggunakan VB 6.0. Peningkatan performa juga dilakukan dengan pergantian komputer dari *operating system* Windows XP menjadi Windows 7.

Setelah semua kedua iterasi selesai dilakukan aplikasi yang telah dibuat akan dipresentasikan untuk diuji, direvisi, dan diperbaiki. Setelah perbaikan selesai dilakukan maka aplikasi berbasis *web* yang telah dibuat akan di-*hosting* pada suatu komputer dan dapat diakses oleh *user* yang menggunakannya melalui alamat IP atau URL *web* komputer *host*-nya.

Tahapan-tahapan yang dilakukan pada iterasi *Agile Model* adalah sebagai berikut.

1. *Planning*, untuk merencanakan dan mempersiapkan peralatan dan perlengkapan yang digunakan untuk membangun sistem.
2. *Requirement Analysis*, untuk menganalisis kebutuhan-kebutuhan berupa software/hardware apa saja yang diperlukan oleh *client* pada sistem.
3. *Designing*, perancangan pembangunan sistem sesuai dengan *requirement* yang sudah didefinisikan pada tahap *requirement analysis*. Pada tahap ini dilakukan desain pada aplikasi yang ingin dibuat, hal ini dapat dibantu dengan cara membuat diagram UML, DFD atau diagram-diagram lainnya untuk membantu memahami sistem secara keseluruhan dan dilakukan desain terhadap tampilan antarmuka (*user interface*)
4. *Building*, merupakan tahap mengimplementasikan desain yang sudah dibuat sebelumnya dengan menulis kode program *coding* untuk mendapatkan luaran yang diinginkan. *Coding* dilakukan dengan menggunakan PHP sebagai server scripting language dan AngularJS sebagai client scripting language.
5. *Testing*, untuk menguji apakah fungsi-fungsi yang dibangun pada tahap *building* sudah berjalan sesuai dengan luaran yang diharapkan. *Testing* dilakukan dengan mendemokan satu per satu fitur-fitur / fungsionalitas yang tersedia pada aplikasi yang sudah dibuat. Apabila luaran yang diharapkan tidak sesuai, maka program akan ditinjau ulang untuk mengetahui letak kesalahan dan diperbaiki. Setelah diperbaiki, program akan melalui proses *testing* ulang sampai mendapatkan luaran yang tepat.

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

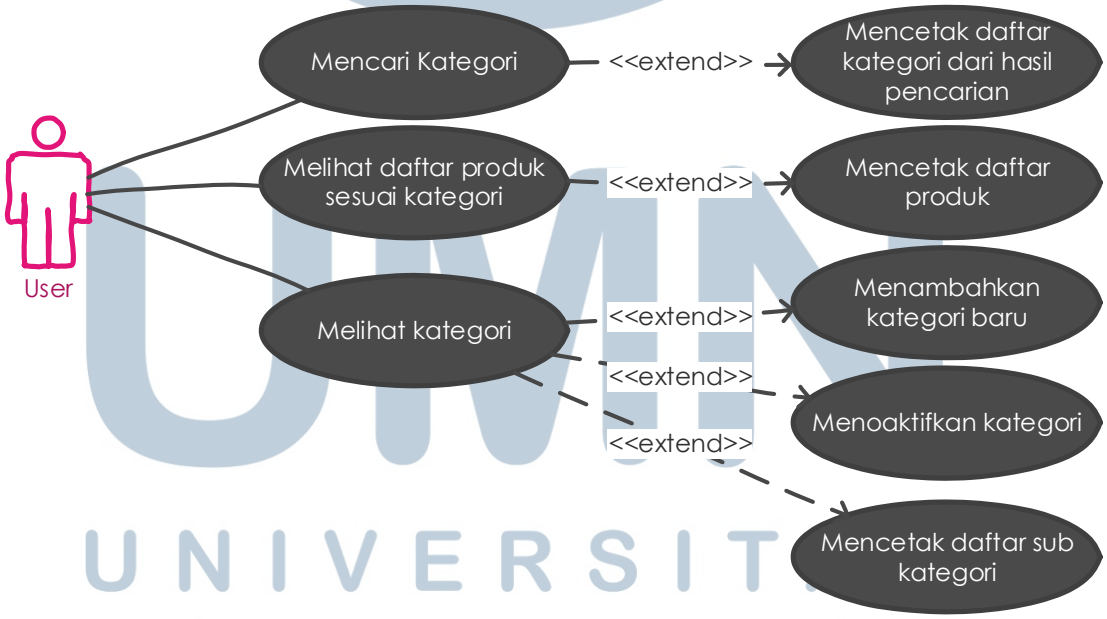
3.3.2 Hasil Pelaksanaan

Hasil pelaksanaan kerja magang berupa Use Case Diagram, Activity Diagram, Class Diagram, Sequence Diagram, kode program (*source code*) dan pengujian dari segala fungsionalistas yang tersedia pada aplikasi yang telah dibuat.

A. Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan interaksi yang dapat dilakukan oleh pengguna terhadap aplikasi yang dikembangkan.

Gambar 3.1 menggambarkan use case diagram untuk Master Category yang berisi semua fungsionalitas yang tersedia pada Master Category.



Gambar 3.1 Use Case Diagram Master Category

UNIVERSITAS MULTIMEDIA NUSANTARA

Tabel 3.2 menjelaskan tentang Use Case untuk mencari kategori yang digunakan oleh user untuk mencari kategori sesuai dengan *keyword* yang telah dimasukan.

Use Case	Mencari kategori
Actor	User
Description	Sebuah event dimana aktor memilih menu Find Category yang terdapat pada navigation bar untuk menampilkan halaman Find Category
Trigger	Aktor memilih menu Find Category yang terdapat pada menu navigasi Master Category
Flow of Event	1.Aktor mengakses halaman utama Master Category. 2.Aplikasi menampilkan halaman utama Master Category 3.Aktor memilih menu find category yang terdapat pada navigation bar 4.Halaman find category kemudian akan ditampilkan
Pre-condition	Aktor telah berada pada menu utama Master Category
Post-condition	Aplikasi akan menampilkan halaman Find Category

Tabel 3.2 Use Case Description "Mencari Kategori"

Tabel 3.3 menjelaskan tentang use case untuk “Mencetak daftar kategori hasil pencarian” yang digunakan oleh user untuk mencetak daftar kategori dari hasil pencarian.

Use Case	Mencetak Daftar Kategori dari Hasil Pencarian
Actor	User
Description	Sebuah event dimana aktor menekan tombol print yang terdapat pada halaman Find Category untuk mencetak daftar kategori hasil pencarian
Trigger	Aktor menekan tombol print yang ada pada halaman Find Category
Flow of Event	1.Aktor mengakses halaman Find Category 2.Aktor menekan tombol print

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

	3.Halaman print preview akan ditampilkan
Pre-condition	Aktor berada pada halaman Find Category dan telah melakukan pencarian kategori
Post-condition	Aplikasi akan menampilkan halaman Print Preview untuk mencetak daftar kategori.

Tabel 3.3 Use Case Description "Mencetak Daftar Kategori dari Hasil Pencarian"

Tabel 3.4 menjelaskan tentang use case “Melihat Daftar Produk Sesuai Kategori” yang digunakan oleh user untuk melihat daftar produk pada kategori yang telah dipilih.

Use Case	Melihat Daftar Produk Sesuai Kategori
Actor	User
Description	Sebuah event dimana aktor memilih menu List Product yang terdapat pada navigation bar untuk menampilkan halaman List Product.
Trigger	Aktor mengakses menu List Product yang ada pada navigation bar Master Category
Flow of Event	1.Aktor mengakses halaman List Product 2.Halaman List Product akan menampilkan daftar produk dalam bentuk tabel.
Pre-condition	Aktor berada pada halaman utama Master Category
Post-condition	Aplikasi akan menampilkan daftar produk yang sesuai dengan kategori yang dipilih pada menu utama Master Category

Tabel 3.4 Use Case Description "Melihat Daftar Produk Sesuai Kategori"

Tabel 3.5 menjelaskan tentang use case “Mencetak Daftar Produk” yang digunakan oleh user untuk mencetak daftar produk pada kategori yang telah dipilih pada menu utama Master Category.

Use Case	Mencetak daftar kategori hasil pencarian
Actor	User
Description	Sebuah event dimana aktor menekan tombol print yang terdapat pada halaman List Product untuk mencetak daftar product
Trigger	Aktor menekan tombol print yang ada pada halaman List Product
Flow of Event	1.Aktor mengakses halaman List Product 2.Aktor menekan tombol print

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

	3.Halaman print preview akan ditampilkan
Pre-condition	Aktor berada pada halaman List Product
Post-condition	Aplikasi akan menampilkan halaman Print Preview untuk mencetak daftar product.

Tabel 3.5 Use Case Description "Mencetak Daftar Produk"

Tabel 3.6 menjelaskan tentang use case “Melihat Kategori” yang digunakan oleh user untuk menampilkan data kategori secara satu persatu dengan menggunakan tombol *next* atau *prev* serta melihat sub kategorinya.

Use Case	Melihat Kategori
Actor	User
Description	Sebuah event dimana aktor membuka halaman Master Category yang merupakan menu utama
Trigger	Aktor mengakses halaman Master Category
Flow of Event	1.Aktor mengakses halaman Master Category 2.Halaman Master Category akan ditampilkan
Pre-condition	Aktor berada pada halaman main navigation bar
Post-condition	Halaman Master Category akan ditampilkan

Tabel 3.6 Use Case Description "Melihat Kategori"

Tabel 3.7 menjelaskan tentang use case “Mencetak Daftar Subkategori” yang digunakan oleh user untuk mencetak daftar sub kategori pada kategori yang telah dipilih pada menu utama Master Category.

Use Case	Mencetak Daftar Subkategori
Actor	User
Description	Sebuah event dimana aktor membuka halaman Master Category yang merupakan menu utama dan menekan tombol print yang terdapat pada tabel sub kategori
Trigger	Aktor menekan tombol print yang ada pada tabel subkategori
Flow of Event	1.Aktor mengakses halaman Master Category 2.Halaman Master Category akan ditampilkan

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

	3.Aktor menekan tombol print yang terdapat pada tabel sub kategori 4.Halaman print preview untuk mencetak akan ditampilkan
Pre-condition	Aktor berada pada halaman Master Category
Post-condition	Halaman Print Preview untuk mencetak daftar sub kategori akan ditampilkan

Tabel 3.7 Use Case Description "Mencetak Daftar Subkategori"

Tabel 3.8 menjelaskan tentang use case “Menambahkan kategori baru” yang digunakan oleh user untuk menambahkan kategori produk baru ke dalam sistem basis data.

Use Case	Menambahkan Kategori Baru
Actor	User
Description	Sebuah event dimana aktor membuka halaman Master Category kemudian menekan tombol add untuk memunculkan formulir Add Category
Trigger	Aktor menekan tombol Add Category
Flow of Event	1.Aktor mengakses halaman Master Category 2.Halaman Master Category akan ditampilkan 3.Aktor menekan tombol Add Category 4.Kotak dialog Add Category akan ditampilkan untuk diisi dengan benar
Pre-condition	Aktor berada pada halaman main navigation bar
Post-condition	Notifikasi aktor berhasil menambah kategori baru

Tabel 3.8 Use Case Description "Menambahkan Kategori Baru"

Tabel 3.9 menjelaskan tentang use case “Menoaktifkan Kategori” yang digunakan oleh user untuk mengubah status kategori yang tadinya aktif menjadi tidak aktif.

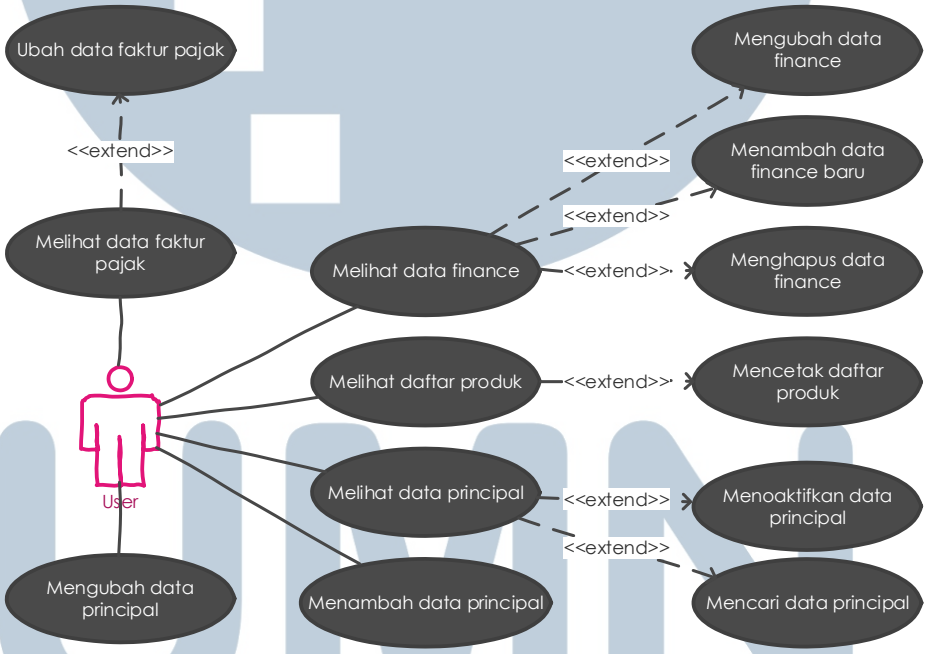
Use Case	Menoaktifkan Kategori
Actor	User
Description	Sebuah event dimana aktor membuka halaman Master Category kemudian menekan tombol Delete
Trigger	Aktor menekan tombol Delete

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

Flow of Event	1.Aktor mengakses halaman Master Category 2.Halaman Master Category akan ditampilkan 3.Aktor menekan tombol Delete
Pre-condition	Aktor berada pada halaman main navigation bar
Post-condition	Notifikasi aktor berhasil mengubah status kategori menjadi tidak aktif

Tabel 3.9 Use Case Description "Menoaktifkan Kategori"

Gambar 3.2 menggambarkan use case diagram untuk Master Principal yang berisi semua fungsionalitas yang tersedia pada Master Principal.



Gambar 3.2 Use Case Diagram Master Principal

Tabel 3.10 menjelaskan tentang use case “Melihat Data Faktur Pajak” yang digunakan oleh user untuk melihat data faktur pajak principal yang dipilih pada menu utama Master Principal

UNIVERSITAS MULTIMEDIA NUSANTARA

Use Case	Melihat Data Faktur Pajak
Actor	User
Description	Sebuah event dimana aktor memilih menu Faktur Pajak yang terdapat pada navigation bar untuk menampilkan halaman Faktur Pajak
Trigger	Aktor memilih menu Faktur Pajak yang terdapat pada menu navigasi
Flow of Event	1.Aktor mengakses halaman utama Master Principal. 2.Aplikasi menampilkan halaman utama Master Principal 3.Aktor memilih menu Faktur Pajak yang terdapat pada navigation bar 4.Halaman Faktur Pajak kemudian akan ditampilkan
Pre-condition	Aktor telah berada pada menu utama Master Principal untuk memilih principal terlebih dahulu
Post-condition	Aplikasi akan menampilkan halaman Faktur Pajak

Tabel 3.10 Use Case Description "Melihat Data Faktur Pajak"

Tabel 3.11 menjelaskan tentang use case “Mengubah Data Faktur Pajak” yang digunakan oleh user untuk mengubah data faktur pajak principal yang dipilih pada menu utama Master Principal.

Use Case	Mengubah Data Faktur Pajak
Actor	User
Description	Sebuah event dimana aktor melakukan pengubahan pada data faktur pajak principal
Trigger	Aktor ingin mengubah data yang terdapat pada halaman Faktur Pajak
Flow of Event	1.Aktor mengakses halaman Faktur Pajak 2.Aplikasi menampilkan halaman Faktur Pajak 3.Aktor mengubah data faktur pajak
Pre-condition	Aktor telah berada pada halaman Faktur Pajak
Post-condition	Aplikasi akan menampilkan notifikasi bahwa data faktur pajak selesai diubah

Tabel 3.11 Use Case Description "Mengubah Data Faktur Pajak"

UNIVERSITAS MULTIMEDIA NUSANTARA

Tabel 3.12 menjelaskan tentang use case “Mengubah Data Principal” yang digunakan oleh user untuk mengubah data principal yang dipilih pada menu utama Master Principal

Use Case	Mengubah Data Principal
Actor	User
Description	Sebuah event dimana aktor melakukan pengubahan pada data principal
Trigger	Aktor melakukan menekan tombol Edit yang ada pada menu navigasi Master Principal
Flow of Event	1.Aktor mengakses halaman utama Master Principal 2.Aplikasi menampilkan halaman Master Principal 3.Aktor melakukan navigasi ke menu Edit Principal 4.Halaman Edit Principal akan ditampilkan
Pre-condition	Aktor telah berada pada halaman utama Master principal
Post-condition	Aplikasi akan menampilkan notifikasi bahwa data principal berhasil diubah

Tabel 3.12 Use Case Description "Mengubah Data Principal"

Tabel 3.13 menjelaskan tentang use case “Menambah Data Principal” yang digunakan oleh user untuk memasukan data principal baru ke *database*.

Use Case	Menambah Data Principal
Actor	User
Description	Sebuah event dimana aktor menambahkan data principal baru
Trigger	Aktor menekan tombol Add pada menu navigasi Master Principal
Flow of Event	1.Aktor mengakses halaman utama Master Principal 2.Aplikasi menampilkan halaman Master Principal 3.Aktor menekan tombol Add pada menu navigasi Master Principal 4.Halaman Add Principal akan ditampilkan
Pre-condition	Aktor telah berada pada halaman utama Master principal
Post-condition	Aplikasi akan menampilkan notifikasi bahwa data principal berhasil ditambah

Tabel 3.13 Use Case Description "Menambah Data Principal"

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

Tabel 3.14 menjelaskan tentang use case “Melihat Data Principal” yang digunakan oleh user untuk melihat data principal secara satu-persatu.

Use Case	Melihat Data Principal
Actor	User
Description	Sebuah event dimana aktor melihat data principal
Trigger	Akto mengakses halaman utama Matser Principal melalui *browser*
Flow of Event	1.Aktor mengakses halaman utama Master Principal 2.Aplikasi menampilkan halaman Master Principal
Pre-condition	Aktor berada pada halaman main navigation bar.
Post-condition	Aplikasi akan menampilkan halaman Master Principal

Tabel 3.14 Use Case Description "Melihat Data Principal"

Tabel 3.15 menjelaskan tentang use case “Menonaktifkan Data Principal” yang digunakan oleh user untuk mengubah status principal menjadi tidak aktif.

Use Case	Menoaktifkan Data Principal
Actor	User
Description	Sebuah event dimana aktor ingin mengubah data principal yang dipilih menjadi tidak aktif
Trigger	Aktor menekan tombol “Non active” yang ada pada menu navigasi
Flow of Event	1.Aktor mengakses halaman utama Master Principal 2.Aplikasi menampilkan halaman Master Principal 3.Aktor menekan tombol “Non active”
Pre-condition	Aktor berada pada halaman Master Principal
Post-condition	Aplikasi akan memunculkan notifikasi data principal telah berhasil dinonaktifkan

Tabel 3.15 Use Case Description "Menoaktifkan Data Principal"

UNIVERSITAS MULTIMEDIA NUSANTARA

Tabel 3.16 menjelaskan tentang use case “Mencari Data Principal” yang digunakan oleh user untuk mencari data principal secara spesifik menggunakan keyword berupa nama *principal* atau kode *principal*.

Use Case	Mencari Data Principal
Actor	User
Description	Sebuah event dimana aktor ingin mencari data principal secara spesifik
Trigger	Aktor menekan tombol Find yang ada pada menu navigasi
Flow of Event	1.Aktor mengakses halaman utama Master Principal 2.Aplikasi menampilkan halaman Master Principal 3.Aktor menekan tombol Find 4.Kotak dialog pencarian akan ditampilkan
Pre-condition	Aktor berada pada halaman Master Principal
Post-condition	Aplikasi akan memunculkan daftar principal yang sesuai dengan masukan kata kunci yang diberikan user

Tabel 3.16 Use Case Description "Mencari Data Principal"

Tabel 3.17 menjelaskan tentang use case “Melihat daftar produk” yang digunakan oleh user untuk melihat daftar produk yang ditawarkan dari principal yang dipilih.

Use Case	Melihat daftar produk
Actor	User
Description	Sebuah event dimana aktor ingin melihat daftar produk yang principal miliki
Trigger	Aktor memilih menu List Product pada menu navigasi
Flow of Event	1.Aktor mengakses halaman utama Master Principal 2.Aplikasi menampilkan halaman Master Principal 3.Aktor memilih menu List Product pada navigation bar 4.Halaman List Product yang berisi daftar produk akan ditampilkan
Pre-condition	Aktor berada pada halaman Master Principal
Post-condition	Aplikasi akan memunculkan halaman List Product

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

Tabel 3.17 Use Case Description "Melihat Daftar Produk"

Tabel 3.18 menjelaskan tentang use case “Mencetak Daftar Produk” yang digunakan oleh user mencetak daftar produk yang ditawarkan principal.

Use Case	Mencetak Daftar Produk
Actor	User
Description	Sebuah event dimana aktor ingin mencetak daftar produk suatu principal
Trigger	Aktor menekan tombol print yang ada pada tabel daftar produk
Flow of Event	1.Aktor berada pada halaman List Product 2.Aktor menekan tombol print yang berada di atas tabel daftar produk
Pre-condition	Aktor berada pada halaman List Product
Post-condition	Aplikasi akan memunculkan halaman print preview untuk mencetak daftar produk

Tabel 3.18 Use Case Description "Mencetak Daftar Produk"

Tabel 3.19 menjelaskan tentang use case “Melihat Data Finance” yang digunakan oleh user untuk melihat data finance principal yang dipilih

Use Case	Melihat Data Finance
Actor	User
Description	Sebuah event dimana aktor ingin melihat data keuangan principal yang dipilih dari menu utama Master Principal
Trigger	Aktor memilih menu Finance yang terdapat pada menu navigasi
Flow of Event	1.Aktor mengakses halaman utama Master Principal 2.Aplikasi menampilkan halaman Master Principal 3.Aktor memilih menu Finance pada menu navigasi 4.Halaman Finance yang berisi data finance principal akan ditampilkan
Pre-condition	Aktor berada pada halaman utama Master Principal untuk memilih principal terlebih dahulu
Post-condition	Aplikasi akan memunculkan halaman Finance untuk menampilkan data finance principal

Tabel 3.19 Use Case Description "Melihat Data Financel"

UNIVERSITAS MULTIMEDIA NUSANTARA

Tabel 3.20 menjelaskan tentang use case “Mengubah Data Finance” yang digunakan oleh user mengubah data finance principal yang dipilih.

Use Case	Mengubah Data Finance
Actor	User
Description	Sebuah event dimana aktor mengubah data yang ada pada halaman Finance
Trigger	Aktor menekan tombol Edit dan mengubah data *finance*
Flow of Event	1.Aktor berada pada halaman Finance 2.Aktor menekan tombol Edit dan mengubah data finance
Pre-condition	Aktor berada pada halaman Finance
Post-condition	Aplikasi akan memunculkan notifikasi bahwa data finance berhasil dilakukan pengubahan

Tabel 3.20 Use Case Description "Mengubah Data Finance"

Tabel 3.21 menjelaskan tentang use case “Menambah Data Finance Baru” yang digunakan oleh user untuk menambahkan data finance

Use Case	Menambah Data Finance Baru
Actor	User
Description	Sebuah event dimana aktor menambahkan data keuangan *principal*
Trigger	Akto meneka tombol Add pada halaman Finance
Flow of Event	1.Aktor berada pada halaman Finance 2.Aktor menekan tombol Add dan melakukan *submit* pada formulir yang telah diisi.
Pre-condition	Aktor berada pada halaman Finance
Post-condition	Aplikasi akan memunculkan notifikasi bahwa data *finance* berhasil ditambahkan

Tabel 3.21 Use Case Description "Menambah Data Finance Baru"

Tabel 3.22 menjelaskan tentang use case “Menghapus Data Finance” yang digunakan oleh user untuk menghapus data finance principal yang dipilih

Use Case	Menghapus Data Finance
Actor	User
Description	Sebuah event dimana aktor menoaktifkan atau menghapus data finance principal

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

Trigger	Aktor Menekan tombol Delete pada halaman Finance
Flow of Event	1.Aktor berada pada halaman Finance 2.Aktor menekan tombol Delete
Pre-condition	Aktor berada pada halaman Finance
Post-condition	Aplikasi akan memunculkan notifikasi bahwa data finance berhasil dihapus

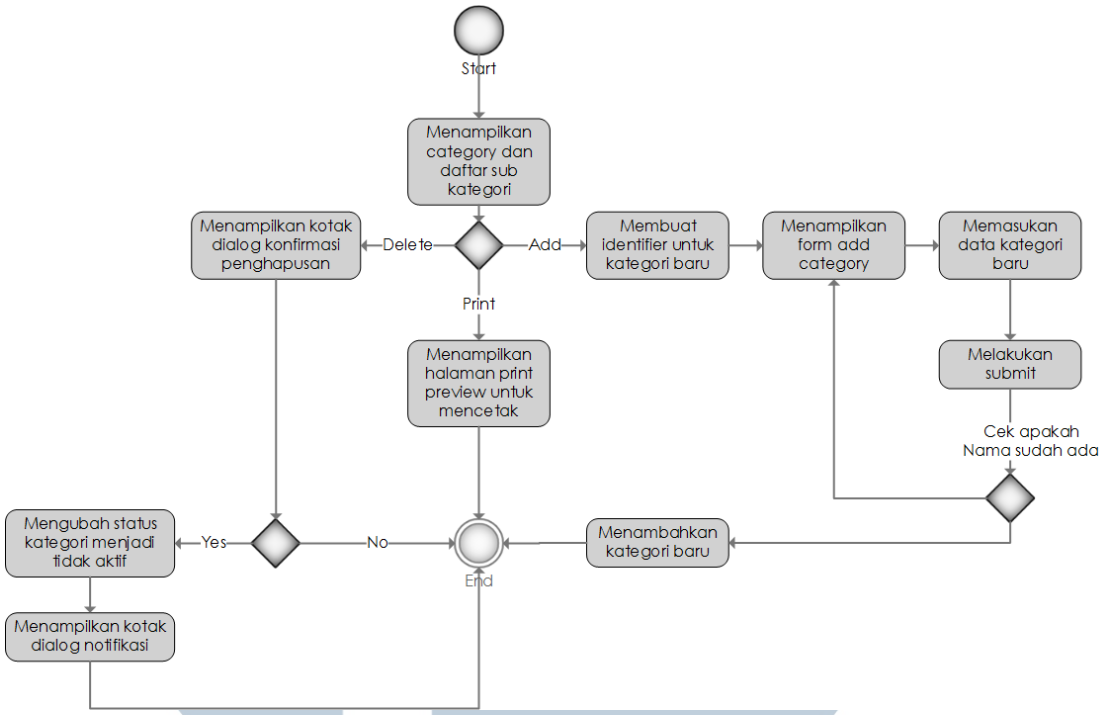
Tabel 3.22 Use Case Description "Menghapus Data Finance"

B. Activity Diagram

Activity diagram digunakan untuk menggambarkan alur dari sebuah aplikasi secara garis besar. Activity diagram yang digambarkan tidak terlalu detail dan hanya menekankan pada keseluruhan alur fungsionalitas dari aplikasi.

Gambar 3.3 menunjukan activity diagram ketika user melihat kategori yang ditampilkan pada halaman utama Master Kategori. Di halaman ini user bisa menambah kategori, delete / menoaktifkan data, dan mencetak sub kategori.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.3 Activity Diagram Melihat Kategori

Ketika user menambah kategori, secara otomatis aplikasi menentukan *identifier* pada kategori baru tersebut kemudian menampilkannya ke formulir penambahan kategori sehingga user hanya perlu memasukan nama kategori saja. Ketika user melakukan submit, aplikasi akan memeriksa apakah nama kategori yang user masukan sudah ada sebelumnya, jika belum ada maka kategori akan ditambahkan ke dalam *database* tapi bila nama kategori sudah ada maka sistem akan memberikan notifikasi bahwa kategori tersebut tidak bisa ditambahkan ke dalam *database*.

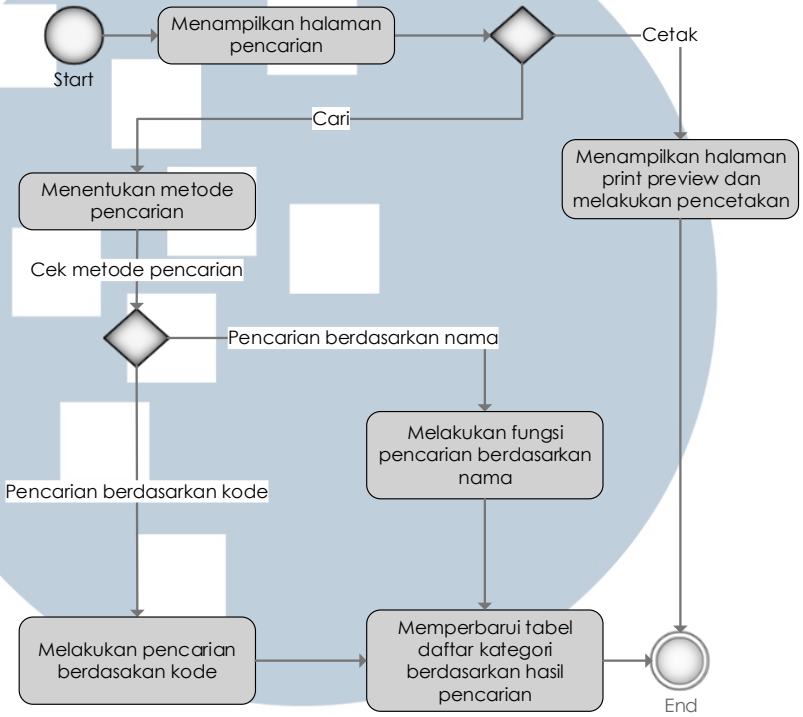
Ketika user menekan tombol delete maka akan muncul kotak dialog konfirmasi penoaktifan data kategori, penoaktifan data hanya dilakukan ketika user memberikan respons positif pada kotak dialog tersebut.

User bisa mencetak daftar sub kategori yang ada pada suatu kategori dengan cara menekan tombol print yang tersedia di tabel sub kategori.

Gambar 3.4 menunjukan activity diagram ketika user melakukan pencarian secara spesifik atau menggunakan kata kunci berupa nama kategori atau kode

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

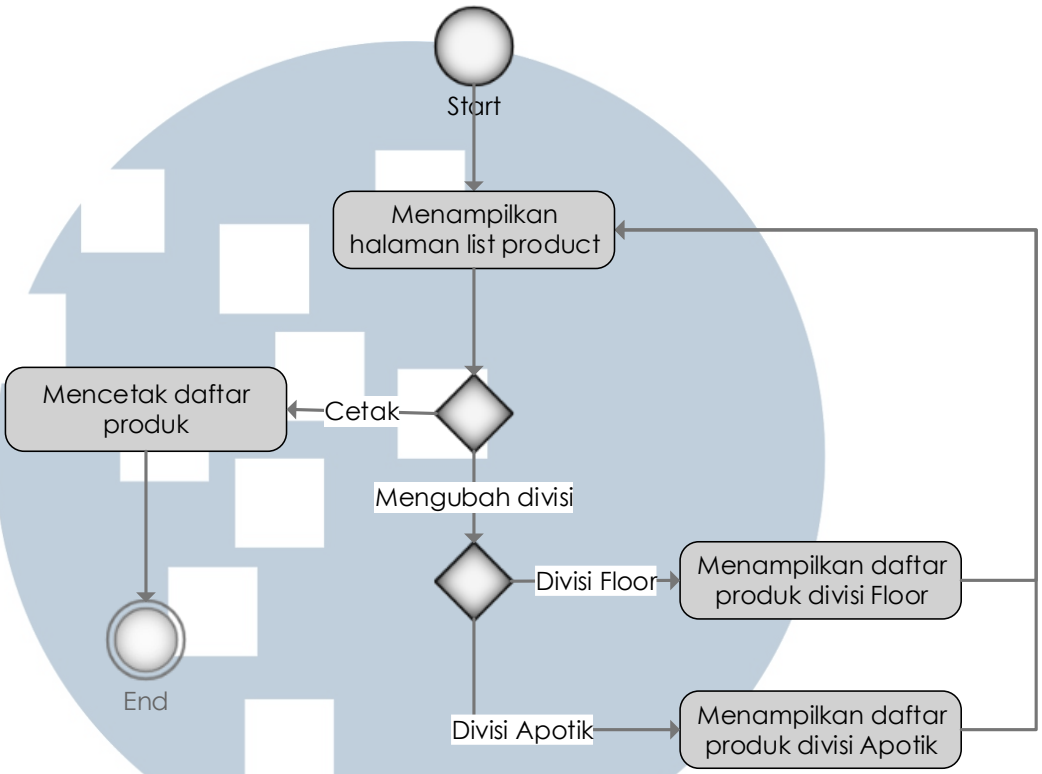
kategori. Di halaman ini user juga bisa melakukan pencetakan terhadap hasil pencarian yang telah dilakukan.



Gambar 3.4 Activity Diagram Mencari Kategori

Gambar 3.5 menunjukan activity diagram ketika user ingin melihat daftar produk dari kategori yang telah dipilih pada menu utama Master Category. Di halaman ini user juga bisa melakukan pencetakan daftar produk kategori tersebut.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



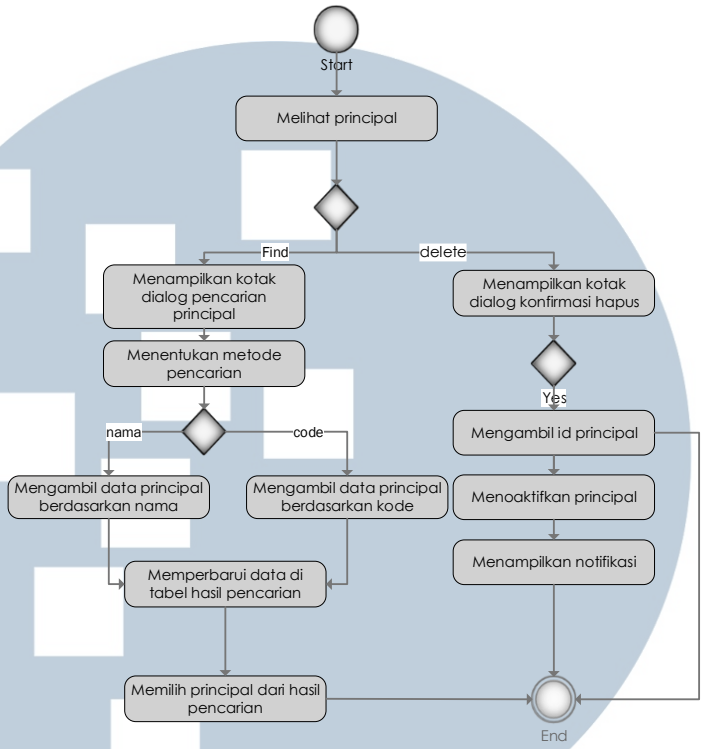
Gambar 3.5 Activity Diagram Melihat Daftar Produk Berdasarkan Kategori

Sebelum dapat menjalankan aktivitas melihat daftar produk, user harus mengakses halaman utama Master Category terlebih dahulu untuk memilih kategori yang ingin ditampilkan daftar produknya.

Pada saat halaman List Product ditampilkan daftar produk yang ditampilkan secara *default* adalah produk-produk untuk divisi apotik. User bisa mengubah divisi melalui menu navigasi yang telah disediakan.

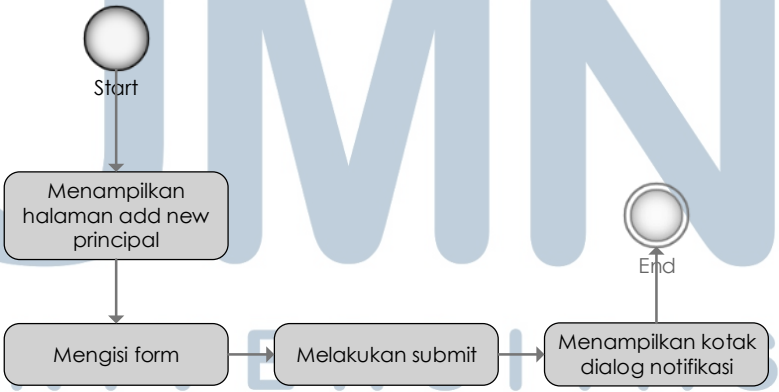
Gambar 3.6 menunjukan activity diagram ketika user ingin melihat data *principal*. Di halaman ini user juga bisa melakukan pencarian principal berdasarkan nama principal atau kode principal, mengubah status principal menjadi tidak aktif

UNIVERSITAS MULTIMEDIA NUSANTARA



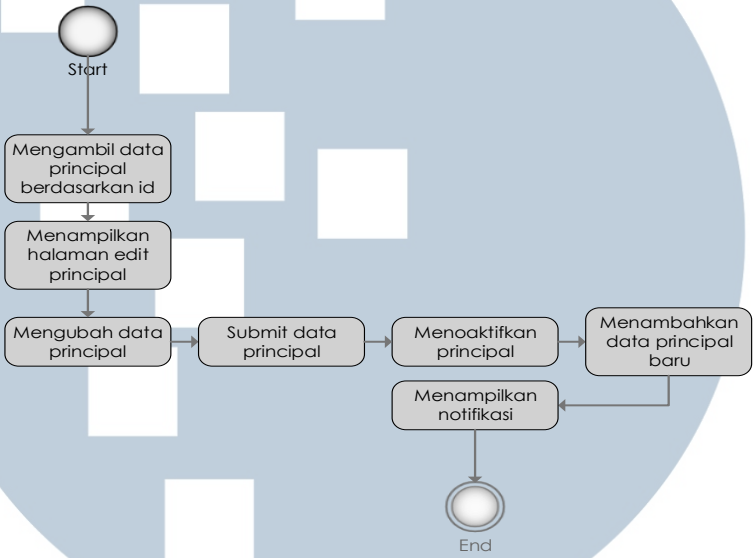
Gambar 3.6 Activity Diagram Melihat Principal

Gambar 3.7 menunjukan activity diagram ketika user ingin menambah data principal. Halaman Add Principal ini bisa diakses melalui tombol navigasi yang berada pada halaman utama Master Principal.



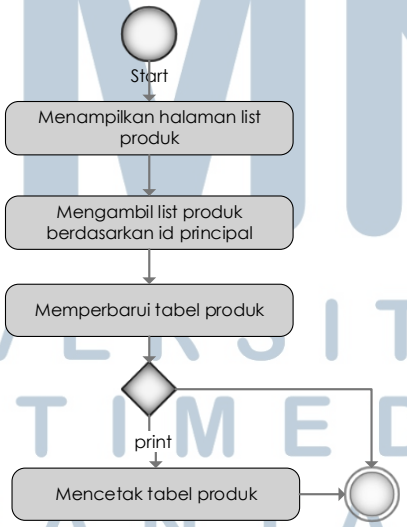
Gambar 3.7 Activity Diagram Menambah Principal

Gambar 3.8 menunjukan activity diagram ketika user ingin mengubah data principal. Halaman Edit Principal ini bisa diakses melalui tombol navigasi yang berada pada halaman utama Master Principal.



Gambar 3.8 Activity Diagram Mengubah Data Principal

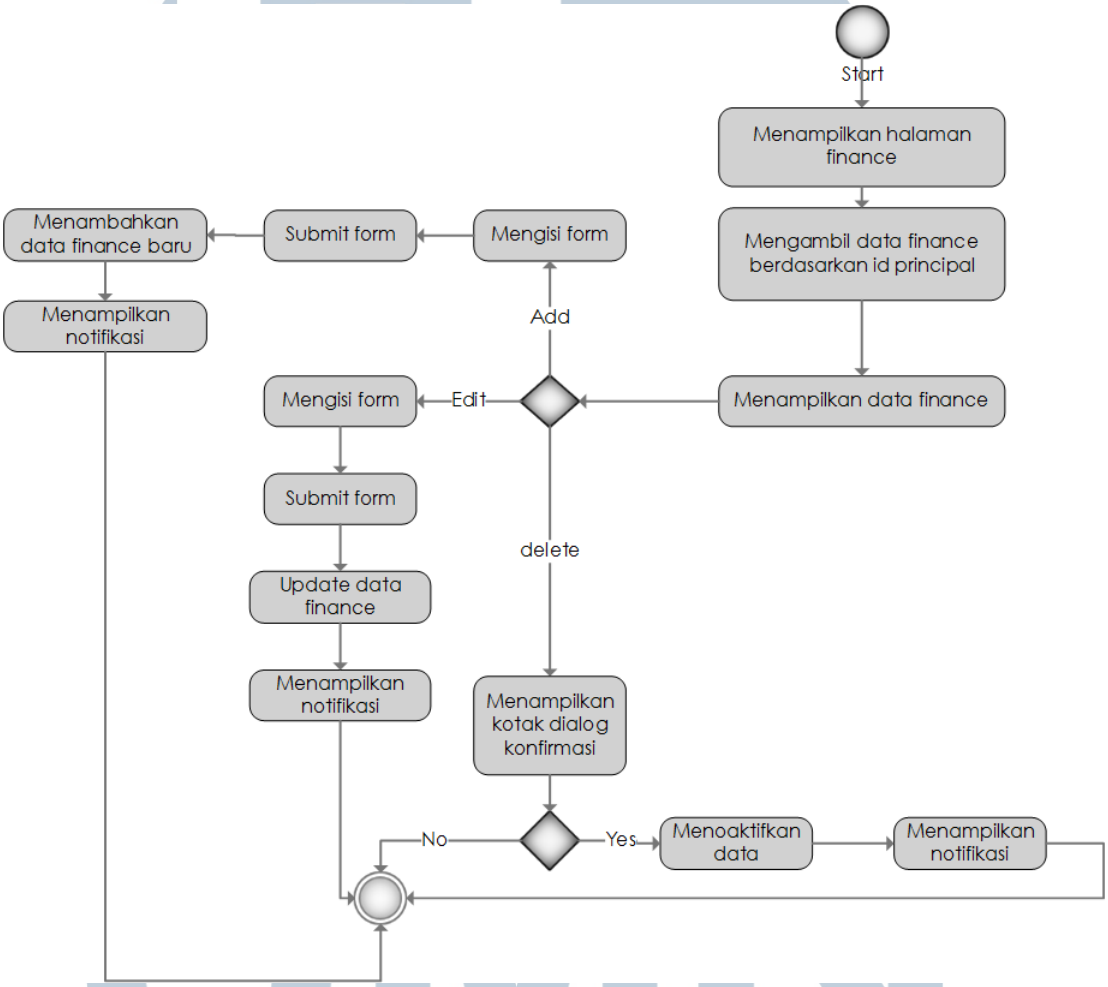
Gambar 3.9 menunjukan activity diagram ketika user ingin melihat daftar produk principal. Halaman List Product Principal ini bisa diakses melalui tombol navigasi yang berada pada halaman utama Master Principal.



Gambar 3.9 Activity Diagram Melihat Daftar Produk Principal

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

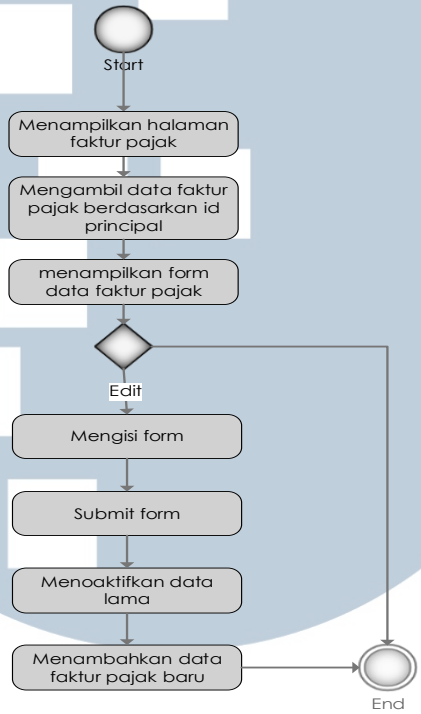
Gambar 3.10 menunjukan activity diagram ketika user mengakses halaman Finance pada menu navigasi Master Principal. Di halaman Finance ini user bisa menambah,mengubah, dan mengubah status data *finance* menjadi tidak aktif.



Gambar 3.10 Activity Diagram Melihat Data Finance Principal

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.11 menunjukan activity diagram ketika user ingin melihat data faktur pajak principal. Di halaman ini user bisa mengubah data faktur pajak yang sudah ada.



Gambar 3.11 Activity Diagram Melihat Data Faktur Pajak Principal

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

C. Sequence Diagram

Perancangan sequence diagram digunakan untuk menunjukan interaksi antar objek dalam modul beserta alur proses yang terjadi dalam modul.

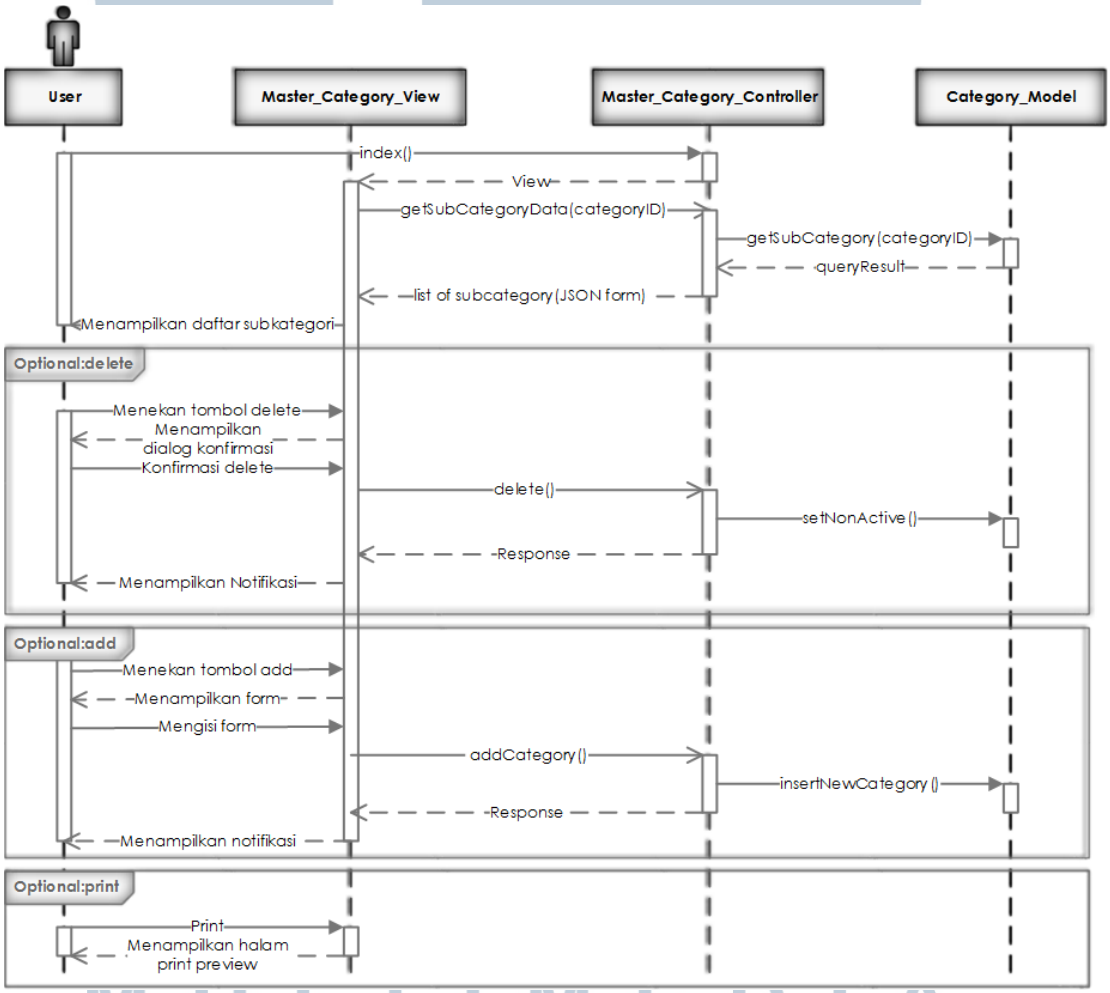
Gambar 3.12 menunjukan Sequence Diagram "Melihat Kategori".Secara garis besar alur dari proses "Melihat Kategori" adalah sebagai berikut:

1. User mengakses URL yang menjalankan fungsi index yang ada pada *Controller* Master Category, dimana fungsi index berfungsi untuk membentuk tampilan halaman *web*.
2. Di dalam view yang sudah dibentuk dari fungsi index pada *controller* Master Category terdapat Javascript yang bertugas untuk melakukan *request* secara asinkronus ke *controller* Master Category untuk meminta data sub kategori melalui pemanggilan fungsi getSubCategoryData.
3. Fungsi getSubCategoryData yang ada pada *controller* Master Category akan memanggil fungsi dengan nama yang sama yang dimiliki model untuk melakukan *query* ke *database* dan mengembalikan hasil *query* tersebut ke *controller*.
4. Sesudah fungsi getSubCategory yang dimiliki *controller* mendapatkan hasil *query* selanjutnya hasil tersebut akan dikonversi menjadi format JSON.
5. Format JSON yang sudah terbentuk akan dikirim ke view dan selanjutnya akan diolah menjadi bentuk tampilan tabel yang menampilkan daftar sub kategori.
6. Jika user menekan tombol Add yang tersedia pada halaman Master Category maka kotak dialog Add Category akan ditampilkan, setelah user memberikan masukan dan melakukan submit maka view akan mengakses fungsi addCategory yang ada pada *controller* Master Category selanjutnya di dalam fungsi tersebut memanggil fungsi yang ada pada Category_Model untuk memasukan data kategori baru ke dalam *database* setelah selesai *controller* akan mengirimkan respons ke view.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

7. Jika user menekan tombol Delete yang tersedia pada halaman Master Category maka kotak dialog konfirmasi akan ditampilkan, apabila memberikan tanggapan positif pada konfirmasi tersebut maka view akan memanggil fungsi delete yang ada pada *controller* Master_Category dan di dalam fungsi tersebut akan memanggil fungsi setNonActive yang dimiliki Category_Model setelah selesai *controller* akan mengirimkan respons ke view kemudian view akan menampilkan kotak dialog notifikasi
8. Jika user menekan tombol print maka Javascript yang ada pada view akan membentuk halaman print preview untuk mencetak daftar sub kategori



Gambar 3.12 Sequence Diagram "Melihat Kategori"

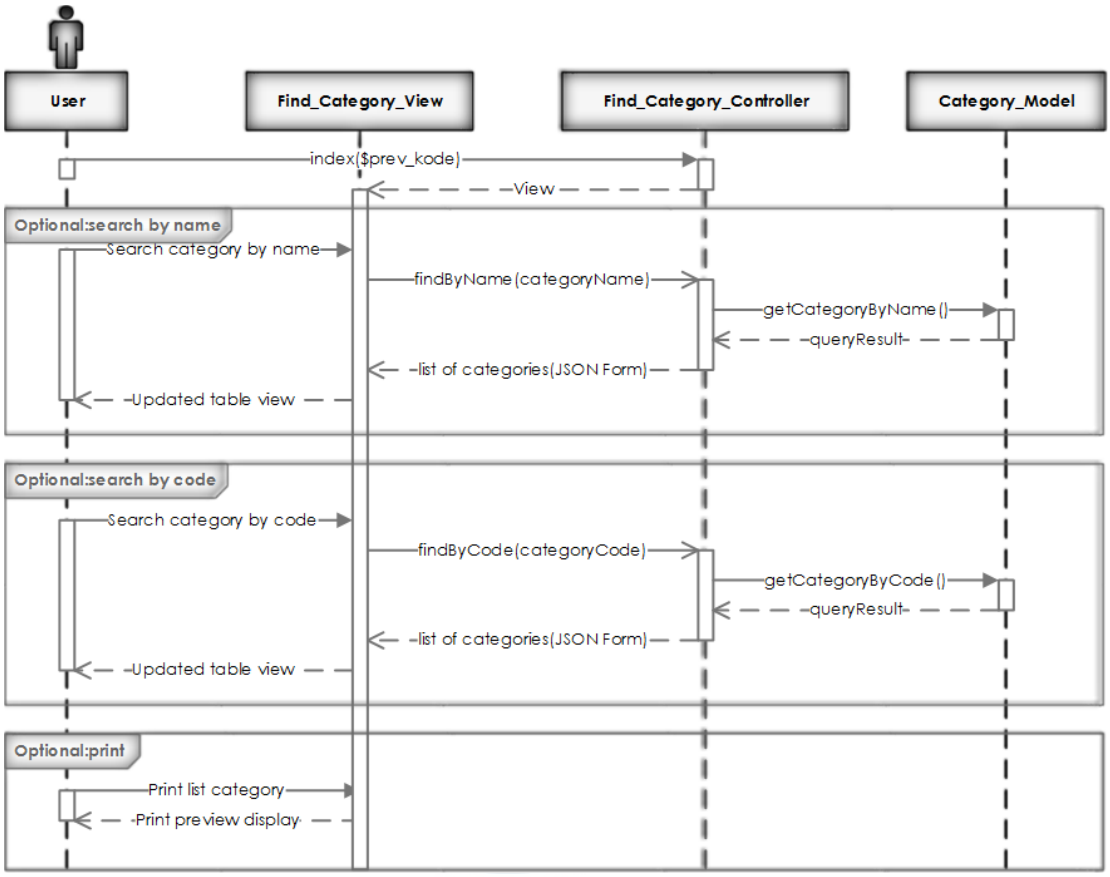
MULTIMEDIA NUSANTARA

Gambar 3.13 menunjukan Sequence Diagram “Mencari Kategori”.Secara garis besar alur dari proses “Mencari Kategori” adalah sebagai berikut:

1. User mengakses URL yang menjalankan fungsi index yang ada pada *controller* Find_Category, dimana fungsi index berfungsi untuk membentuk tampilan halaman *web* untuk halaman Find_Category_View.
2. Setelah view terbentuk user bisa melakukan pencarian berdasarkan nama atau kode kategori
3. Jika user memilih untuk melakukan pencarian berdasarkan nama, maka Javascript yang ada pada Find_Category_View akan melakukan request secara asinkronus untuk mengakses fungsi findByName yang ada pada *controller* Find_Category, selanjutnya fungsi tersebut akan memanggil fungsi yang dimiliki oleh Category_Model untuk melakukan *query*, hasil *query* akan dikirim kembali ke *controller* dan selanjutnya dikonversi menjadi bentuk JSON.
4. JSON yang sudah terbentuk akan dikirim ke view, dan sesudah diterima oleh Find_Category_View akan diolah menjadi bentuk tampilan daftar tabel.
5. Proses yang sama juga terjadi ketika metode pencarian yang dilakukan berdasarkan kode kategori, bedanya fungsi *controller* yang dipanggil oleh JavaScript adalah findByCode
6. Jika user menekan tombol print yang ada pada tabel pencarian maka Javascript pada Find_Category_View akan membentuk halaman print preview untuk mencetak hasil pencarian yang dilakukan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017



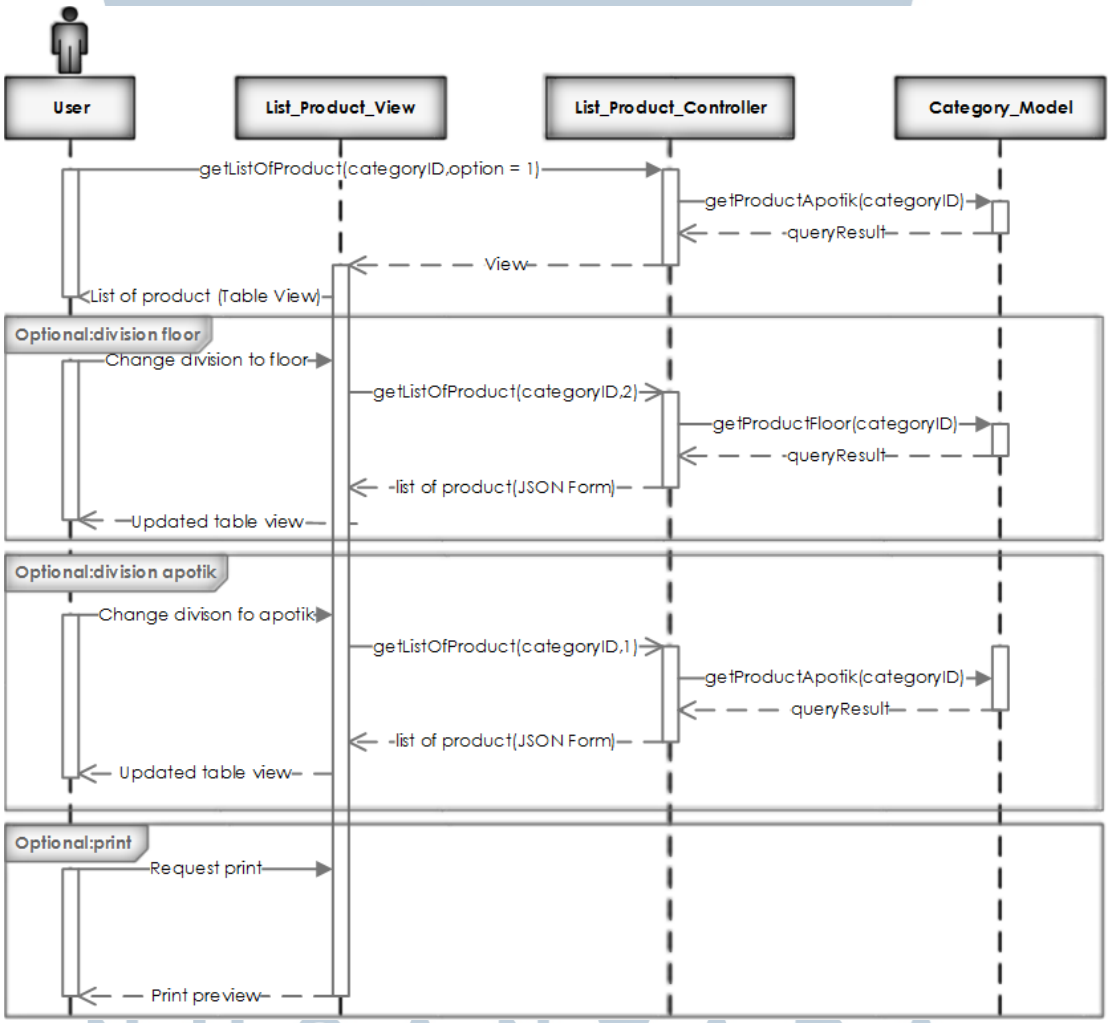
Gambar 3.13 Sequence Diagram "Mencari Kategori"

Gambar 3.14 menunjukan Sequence Diagram “Melihat Daftar Produk”.Secara garis besar alur dari proses “Melihat Daftar Produk” adalah sebagai berikut:

1. User mengakses URL yang menjalankan fungsi getListOfProduct yang ada pada *controller* List_Product, dimana fungsi tersebut berfungsi untuk mengambil daftar produk dan membentuk tampilan halaman *web* untuk halaman List_Product_View.
2. Secara default fungsi getListOfProduct mengambil daftar produk untuk divisi apotik, user bisa mengubah divisi melalui menu navigasi yang ada di halaman tersebut.
3. Jika user mengubah divisi, Javascript yang ada pada view akan melakukan *request* secara asinkronus untuk mengakses fungsi getListOfProduct dengan parameter bernilai 1 apabila divisi yang dipilih adalah Apotik dan nilai 2

apabila divisi yang dipilih adalah Floor. Fungsi pada *controller* tersebut akan memanggil fungsi yang ada pada Category_Model untuk melakukan *query*, hasil *query* dari Category_Model akan dikembalikan ke List_Product_*Controller* dan mengubahnya menjadi format JSON untuk dikirim ke List_Product-View.

4. Javascript pada List_Product_View akan menerima data dengan format JSON dan memperbarui tabel daftar produk.
5. Jika user menekan tombol print, Javascript yang ada pada List_Product_View akan membentuk halaman print preview yang digunakan untuk mencetak daftar produk



Gambar 3.14 Sequence Diagram "Melihat Daftar Produk"

NUSANTARA

Gambar 3.15 menunjukan Sequence Diagram “Melihat Principal”.Secara garis besar alur dari proses “Melihat Principal” adalah sebagai berikut:

1. User mengakses URL yang menjalankan fungsi index yang ada pada *controller* Master_Principal, dimana fungsi tersebut berfungsi menampilkan halaman utama Master_Principal.
2. Setelah Master_Principal_View terbentuk, maka Javascript yang ada pada view tersebut akan menggunakan fungsi get_PrincipalData yang dimiliki oleh prinFactory.
3. Fungsi get_PrincipalData akan mengakses URL untuk menjalankan fungsi getPrincipalData yang dimiliki oleh Master_Principal_*Controller*, fungsi *controller* tersebut memanggil fungsi yang ada pada Principal_Model untuk melakukan *query*. Hasil *query* yang didapat dari Principal_Model akan dikirim ke *controller* untuk diubah formatnya menjadi JSON untuk dikirim ke Master_Principal_View.
4. Setelah Master_Principal_View menerima JSON maka Javascript yang pada pada view tersebut menggunakan fungsi yang dimiliki prinFactory untuk mendapatkan daftar contact person dari principal tersebut.
5. Proses yang dilakukan hampir sama sebelumnya dimana prinFactory mengakses fungsi *controller* dan *controller* menggunakan fungsi pada model untuk melakukan *query*, mengubah formatnya menjadi JSON untuk dikirim ke view.
6. Setelah data contact person diterima dalam bentuk JSON, Master_Principal_View akan mengolahnya menjadi sebuah tampilan yang menggambarkan informasi tentang profil principal.
7. Apabila user menekan tombol delete maka Master_Principal_View akan menggunakan fungsi deactive_principal yang dimiliki Master_Principal_*Controller* dan *controller* tersebut akan menggunakan fungsi pada Principal_Model untuk melakukan operasi *database* untuk mengubah status principal menjadi tidak aktif setelah selesai *controller* akan

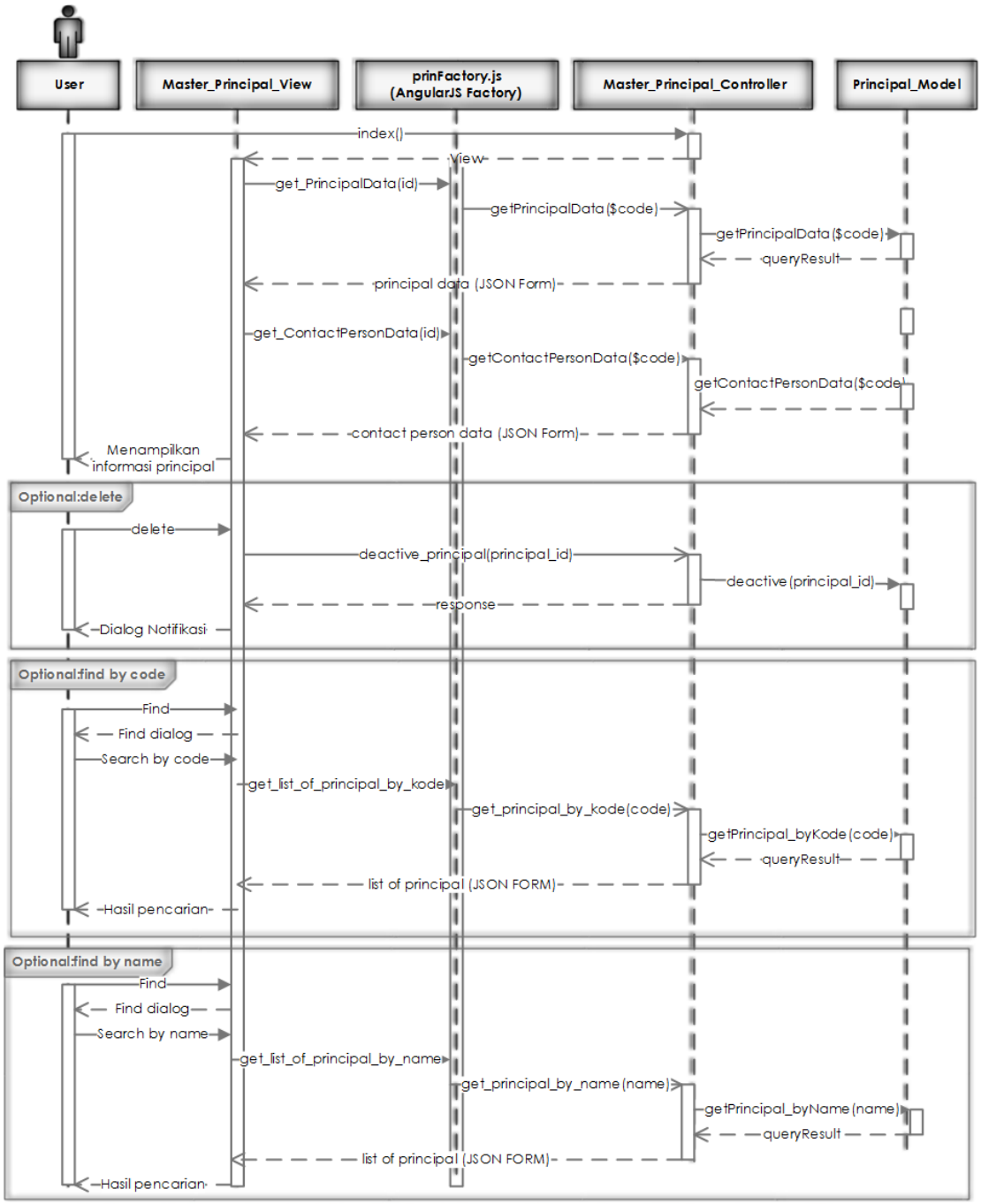
Rancang bangun aplikasi..., Michael, FTI UMN, 2017

mengirimkan respons kepada view. Selanjutnya view akan menampilkan hasil respon tersebut dalam bentuk kotak dialog notifikasi.

8. Apabila user menekan tombol find, maka kotak dialog find akan ditampilkan, user bisa memilih metode pencarian principal dengan menggunakan kode atau nama principal. Setelah menentukan metode pencarian dan memasukan kata kunci, view akan mengakses fungsi get_list_of_principal_by_kode apabila metode pencarian menggunakan kode dan fungsi get_list_of_principal_by_nama apabila metode pencarian menggunakan nama. Kedua fungsi tersebut akan mengakses fungsi yang ada pada Master_Principal_*Controller* dan *controller* akan menggunakan fungsi pada model untuk mendapatkan hasil *query* untuk dikonversi menjadi JSON dan dikirimkan ke view.
9. Setelah view mendapatkan data dalam format JSON maka tabel hasil pencarian yang ada di kotak dialog akan diperbarui.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

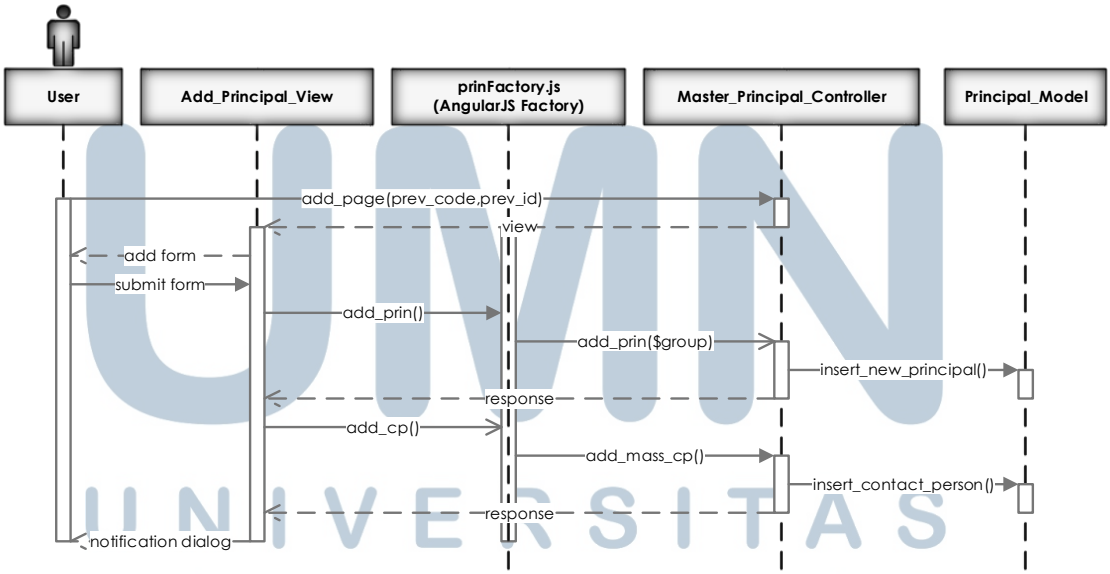


Gambar 3.15 Sequence Diagram "Melihat Principal"

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.16 menunjukan Sequence Diagram "Menambah Data Principal". Secara garis besar alur dari proses "Menambah Data Principal" adalah sebagai berikut:

1. User mengakses URL yang menjalankan fungsi add_page yang ada pada *controller* Master_Principal, dimana fungsi tersebut berfungsi menampilkan halaman Add_Principal_View.
2. Sesudah view ditampilkan, user mengisi form dan melakukan submit
3. Objek prinFactory yang ada pada Javascript akan menggunakan fungsi add_prin untuk mengakses fungsi yang ada di Master_Principal_*Controller* dimana fungsi tersebut mengambil input yang diberikan user dalam bentuk JSON.
4. JSON yang diterima oleh *controller* akan dilakukan proses *decode* dan menggunakan fungsi pada Principal_Model untuk menambahkan ke *database*.
5. Setelah proses penambahan ke *database* selesai *controller* akan mengirim respons ke view dan ditampilkan ke user dalam bentuk kotak dialog



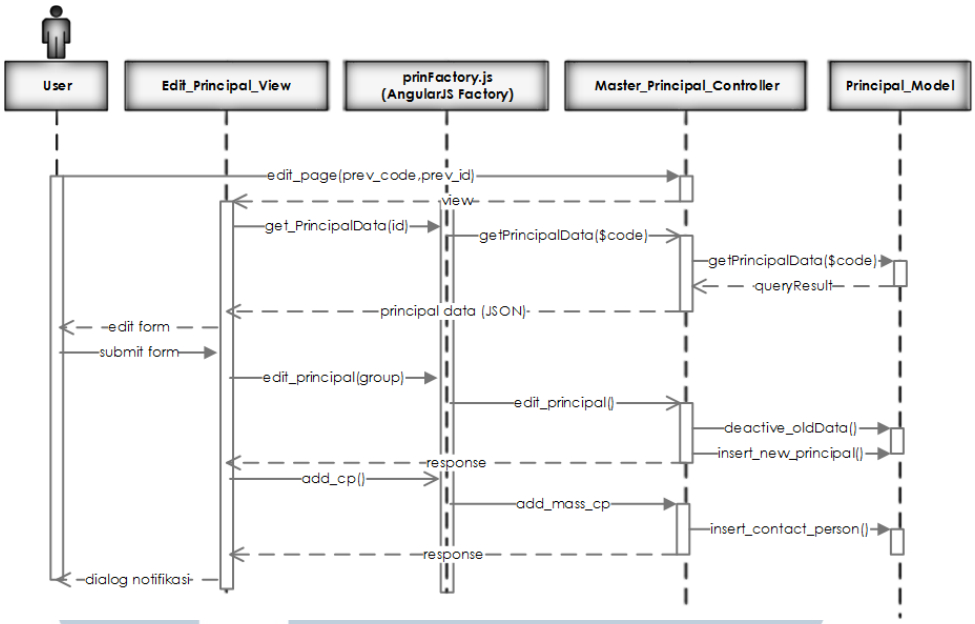
Gambar 3.16 Sequence Diagram "Menambah Data Principal"

Gambar 3.17 menunjukan Sequence Diagram “Mengubah Data Principal”. Secara garis besar alur dari proses “Mengubah Data Principal” adalah sebagai berikut:

1. User mengakses URL yang menjalankan fungsi edit_page yang ada pada *controller* Master_Principal, dimana fungsi tersebut berfungsi membentuk halaman Edit_Principal_View.
2. Setelah Edit_Principal_View terbentuk, Javascript yang ada pada view tersebut akan menggunakan fungsi get_PrincipalData yang dimiliki oleh prinFactory.
3. Fungsi tersebut berfungsi untuk mengambil data principal dalam bentuk JSON untuk mengisi form edit principal secara *default*.
4. Ketika user selesai mengubah data principal, Edit_Principal_View akan menggunakan fungsi edit_principal yang dimiliki prinFactory untuk mengakses fungsi edit_principal pada *controller*. Fungsi edit_principal tersebut akan menggunakan dua buah fungsi pada Principal_Model yang tujuannya untuk mengubah status data principal sebelum diubah menjadi tidak aktif dan menambahkan data principal baru.
5. Setelah proses selesai, *controller* akan mengirimkan respon ke view dan view akan mengolah respon tersebut menjadi sebuah kotak dialog notifikasi yang ditampilkan ke user.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017



Gambar 3.17 Sequence Diagram "Mengubah Data Principal"

Gambar 3.18 menunjukan Sequence Diagram "Melihat Data Finance Principal". Secara garis besar alur dari proses "Melihat Data Finance Principal" adalah sebagai berikut:

1. User mengakses URL yang menjalankan fungsi finance yang ada pada *controller* Master_Principal, dimana fungsi tersebut berfungsi membentuk halaman Finance_View.
2. Javascript yang ada pada Finance_View akan menggunakan fungsi milik prinFactory untuk mengambil data finance yang sudah ada pada *database* dalam bentuk JSON melalui Master_Principal_*Controller*.
3. Master_Principal_*Controller* akan menggunakan fungsi pada Principal_Model untuk mengambil data finance yang ada pada *database* kemudian hasilnya dijadikan dalam format JSON.
4. Data finance yang sudah menjadi format JSON akan dikirim ke Finance_View untuk ditampilkan.
5. Sesudah data finance ditampilkan user bisa menekan tombol edit untuk mengubah data yang sudah ada, menekan tombol add untuk menambah data

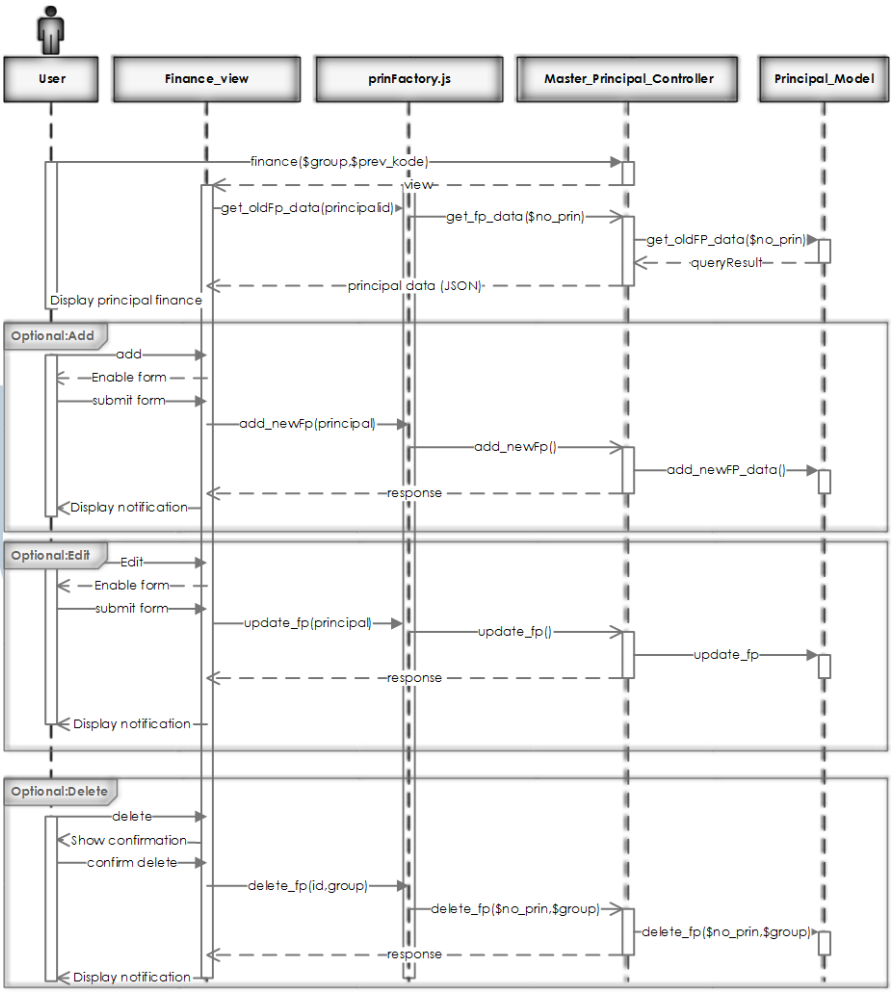
Rancang bangun aplikasi..., Michael, FTI UMN, 2017

finance dan menekan tombol delete untuk mengubah status data finance menjadi tidak aktif.

6. Jika user menekan tombol edit form yang tadinya dalam status disabled (tidak dapat diubah) menjadi enabled (bisa melakukan pengubahan data).
7. Ketika user menekan tombol save, fungsi update_fp yang ada pada prinFactory akan dipanggil untuk mengirim masukan dari user dalam bentuk JSON ke fungsi update yang ada pada Master_Principal_*Controller*.
8. Fungsi update yang ada pada *controller* akan melakukan *decode* pada JSON yang diterima dan selanjutnya menggunakan fungsi pada Principal_Model untuk melakukan update ke *database*. Setelah selesai *controller* akan memberikan respon ke view dan view menampilkan respon tersebut dalam bentuk kotak dialog notifikasi.
9. Jika user menekan tombol add, fungsi add_newFp pada prinFactory akan dipanggil untuk mengirim masukan data finance baru dalam bentuk JSON ke fungsi yang ada pada Master_Principal_*Controller* selanjutnya *controller* tersebut akan menggunakan fungsi pada model untuk memasukan data finance baru ke *database*. Setelah selesai maka *controller* akan mengirimkan respons ke view dan view menampilkan respons tersebut dalam bentuk kotak dialog.
10. Jika user menekan tombol delete maka fungsi delete_fp milik prinFactory dipanggil untuk mengakses fungsi yang ada di Master_Principal_*Controller*. Fungsi pada *controller* tersebut akan menggunakan fungsi yang dimiliki Principal_Model untuk mengubah status data finance yang dipilih menjadi tidak aktif. Setelah selesai maka *controller* akan mengirimkan respon ke view dan respon tersebut ditampilkan dalam bentuk kotak dialog notifikasi.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

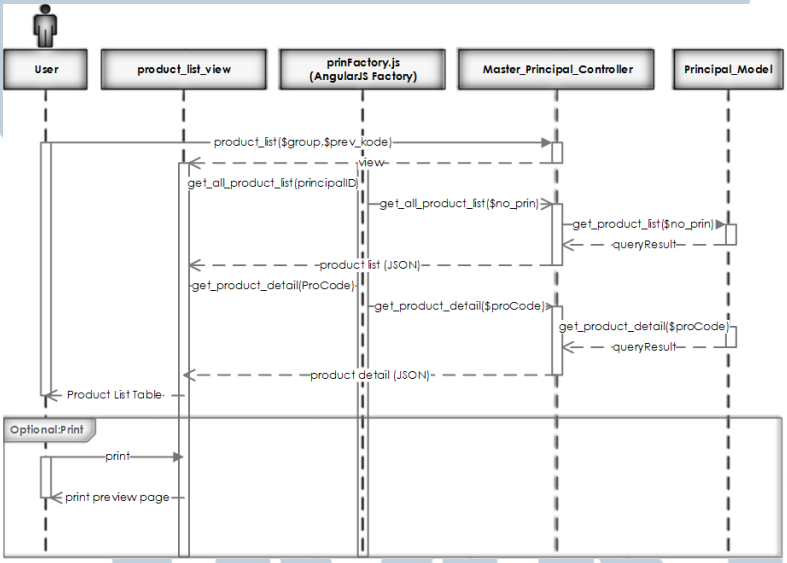


Gambar 3.18 "Melihat Data Finance Principal"

Gambar 3.19 menunjukan Sequence Diagram "Melihat Daftar Produk Principal". Secara garis besar alur dari proses "Melihat Daftar Produk Principal" adalah sebagai berikut:

1. User mengakses URL yang menjalankan fungsi product_list yang ada pada *controller* Master_Principal, dimana fungsi tersebut berfungsi membentuk halaman Product_List_View.
2. Javascript yang ada pada view tersebut akan memanggil fungsi get_all_product_list yang dimiliki prinFactory untuk mengakses fungsi pada Master_Principal_*Controller* yang mengembalikan daftar produk dalam bentuk JSON.

3. Setelah daftar produk dikirim dari *controller* ke view, Javascript pada List_Product_View memanggil fungsi get_product_detail yang dimiliki prinFactory tujuannya adalah untuk mendapatkan detail dari setiap produk dengan mengakses fungsi yang ada pada Master_Principal_*Controller*.
4. Master_Principal_*Controller* akan mengakses fungsi pada Principal_Model untuk memperoleh hasil *query* dan melakukan konversi menjadi bentuk JSON.
5. Setelah data detail produk diterima oleh view, Javascript pada view tersebut akan melakukan perhitungan dan menampilkan daftar produk beserta detailnya dalam bentuk tabel.



Gambar 3.19 Sequence Diagram "Melihat Daftar Produk"

Gambar 3.20 menunjukan Sequence Diagram "Melihat Data Faktur Pajak". Secara garis besar alur dari proses "Melihat Data Faktur Pajak" adalah sebagai berikut:

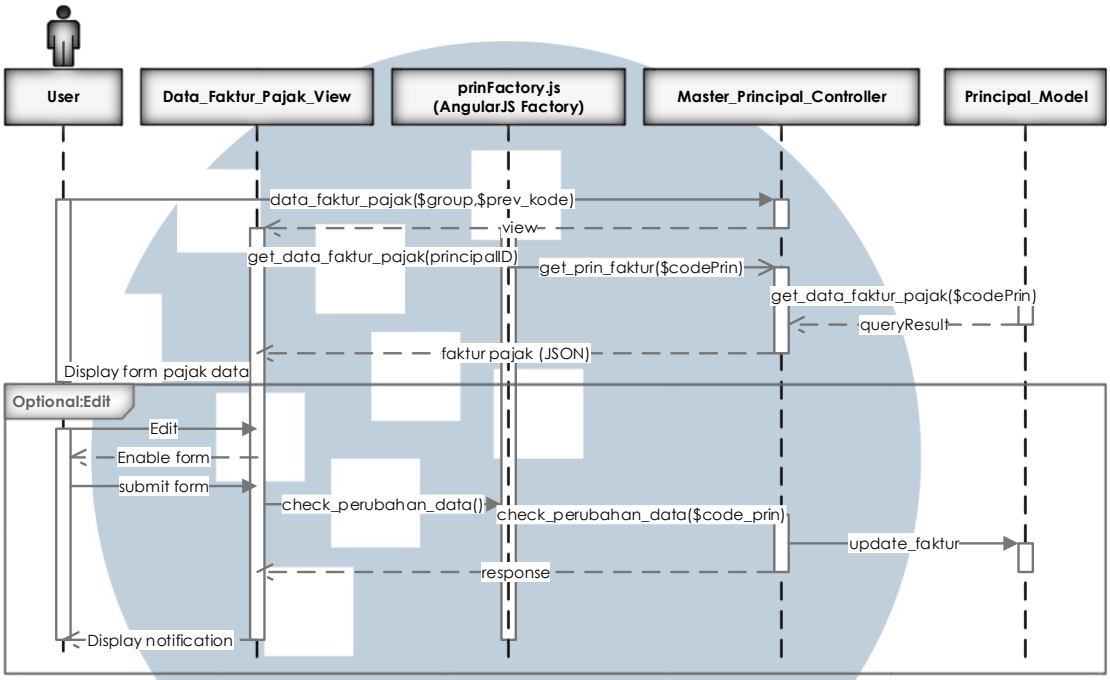
1. User mengakses URL yang menjalankan fungsi data_faktur_pajak yang ada pada Master_Principal_*Controller*, dimana fungsi tersebut berfungsi membentuk halaman Data_Faktur_Pajak_View.

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

2. Javascript pada view yang telah terbentuk akan menjalankan fungsi get_data_faktur_pajak yang dimiliki prinFactory untuk mengakses fungsi get_prin_faktur yang ada pada Master_Principal_*Controller*.
3. Fungsi *controller* yang diakses akan menggunakan Principal_Model untuk mendapatkan data faktur pajak principal yang dipilih, hasil *query* dari model tersebut akan dikirimkan ke *controller* untuk diubah formatnya menjadi JSON.
4. Setelah format data dikonversi dalam bentuk JSON, formulir yang ada pada halaman faktur pajak akan berisi data faktur pajak principal.
5. Jika user menekan tombol edit, maka form yang tadinya tidak bisa diisi oleh user menjadi bisa untuk diubah.
6. Ketika user melakukan submit pada form yang sudah diisi, prinFactory akan memanggil fungsi check_perubahan_data untuk mengakses fungsi *controller* yang berguna untuk mengecek apakah data faktur pajak yang diubah mengalami perubahan atau tidak.
7. Jika data mengalami perubahan maka dilakukan update data faktur pajak yang ada pada *database* melalui pemanggilan fungsi yang terdapat pada Principal_Model, setelah selesai *controller* akan mengembalikan respons yang menandakan apakah update data faktur pajak yang dilakukan berhasil atau tidak kepada Data_Faktur_Pajak_View.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017



Gambar 3.20 Sequence Diagram "Melihat Data Faktur Pajak"

D. Class Diagram

Perancangan class diagram digunakan untuk menunjukan hubungan antar class dalam tiap modul. Setiap modul akan memiliki beberapa jenis class yaitu

1. *Controller*: berisi fungsi-fungsi yang digunakan untuk menyediakan semua fungsionalitas suatu aplikasi.
2. Model: berisi fungsi-fungsi yang digunakan untuk melakukan operasi ke *database* (insert,update,delete,select) dan selalu mengembalikan nilai berupa hasil *query*.
3. View yang terdiri dari objek-objek Javascript yang digunakan untuk menampung data yang didapat dari pemanggilan *controller*.
4. Factory yang merupakan fungsi yang mengembalikan sebuah objek yang terdapat pada modul AngularJS. Tujuannya dibuatnya factory adalah untuk mengakses semua fungsi yang ada pada *controller*.

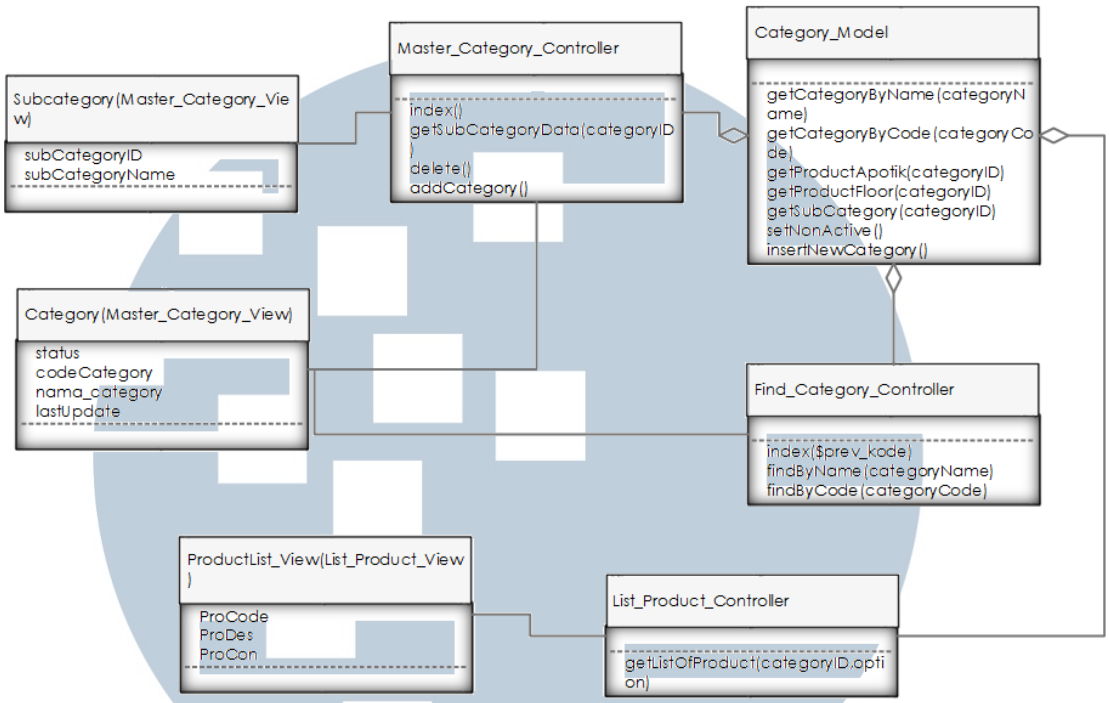
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.21 menunjukan Class Diagram untuk aplikasi Master Category. Berikut penjelasan mengenai fungsi pada setiap class:

1. Class SubCategory adalah class yang terdapat pada Master_Category_View yang berfungsi untuk menampung data subkategori yang dikirimkan oleh *controller* dalam bentuk JSON.
2. Class Category adalah class yang terdapat pada Master_Category_View yang berfungsi untuk menampung data Category yang dikirimkan oleh *controller* dalam bentuk JSON.
3. Class ProductList adalah class yang terdapat pada Product_List_View yang berfungsi untuk menampung data produk yang dikirimkan oleh *controller* dalam bentuk JSON.
4. Class Master_Category_*Controller* adalah class yang diturunkan dari CI_*Controller* yang digunakan untuk menyediakan semua fungsionalitas pada halaman Master_Category_View (halaman utama)
5. Class List_Product_*Controller* adalah class yang diturunkan dari CI_*Controller* yang digunakan untuk menyediakan semua fungsionalitas dari halaman List_Peoduct_*Controller*.
6. Class Find_Category_*Controller* adalah class yang diturunkan dari CI_*Controller* yang digunakan untuk menyediakan semua fungsionalitas dari halaman Find_Category_View.
7. Class Category_Model adalah class yang diturunkan dari CI_Model yang digunakan untuk melakukan operasi *database*. Class ini selalu digunakan oleh Master_Category_*Controller*, Find_Category_*Controller* dan List_Product_*Controller* .

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017



Gambar 3.21 Class Diagram Master Category

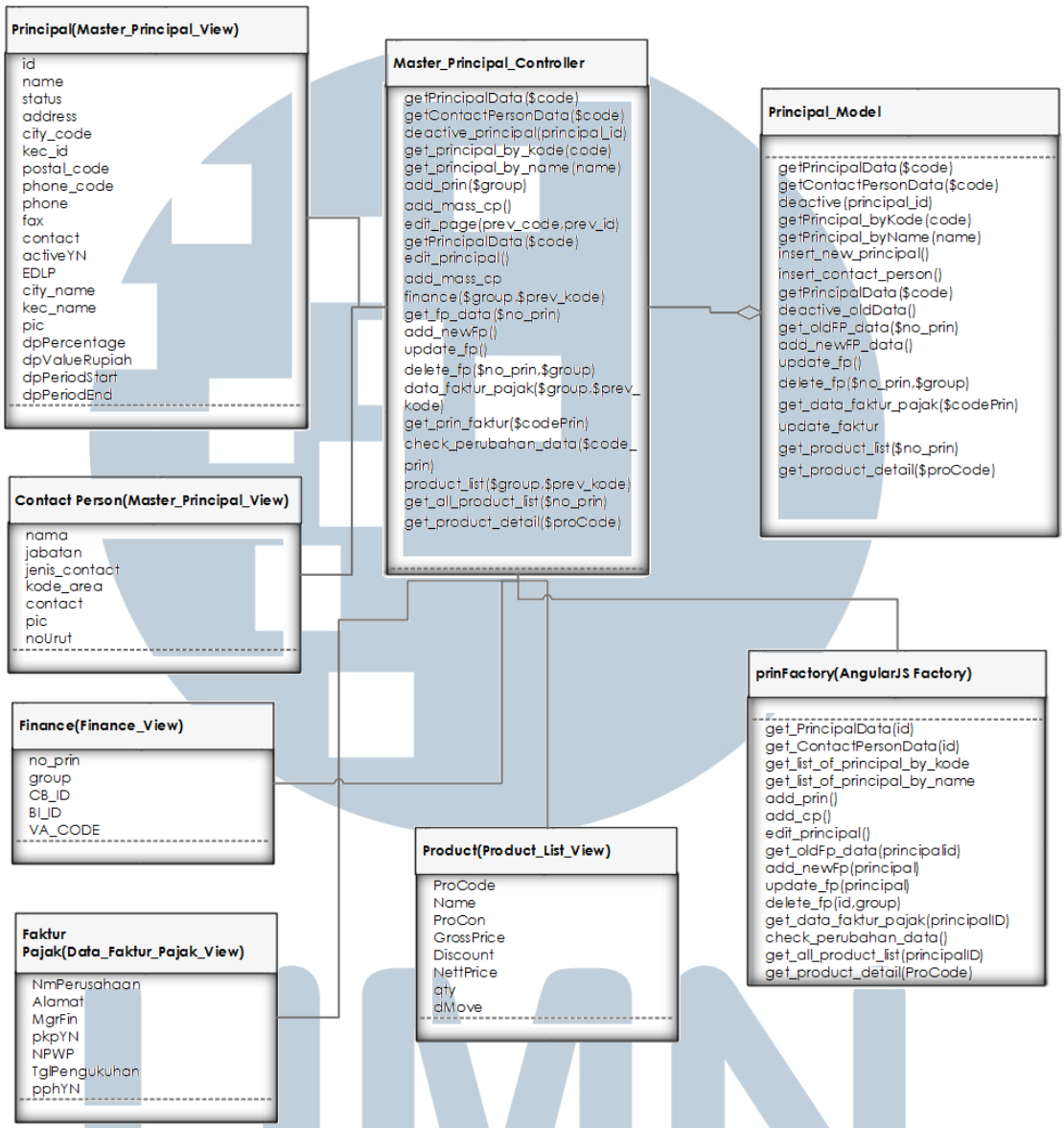
Gambar 3.22 menunjukan Class Diagram untuk aplikasi Master Principal. Berikut penjelasan mengenai fungsi pada setiap class:

1. Class Principal adalah class yang terdapat pada halaman Add_Principal_View, Edit_Principal_View, dan halaman utama Master_Principal, fungsi dari class ini adalah untuk menampung data principal yang nantinya akan dikonversi menjadi JSON atau menerima data principal dalam bentuk JSON.
2. Class Contact Person adalah class yang terdapat pada halaman Add_Principal_View, Edit_Principal_View, dan halaman utama Master_Principal, fungsi dari class ini adalah untuk menampung data contact person yang nantinya akan dikonversi menjadi JSON atau menerima data contact person dalam bentuk JSON.
3. Class Finance adalah class yang terdapat pada halaman Finance fungsi dari class ini adalah untuk menampung data finance yang nantinya akan dikonversi menjadi JSON atau menerima finance dalam bentuk JSON.

4. Class Faktur Pajak adalah class yang terdapat pada halaman Faktur_Pajak_View fungsi dari class ini adalah untuk menampung data faktur pajak yang nantinya akan dikonversi menjadi JSON atau menerima data faktur pajak dalam bentuk JSON.
5. Class Product adalah class yang terdapat pada halaman List_Product_View fungsi dari class ini adalah untuk menampung data product yang nantinya akan dikonversi menjadi JSON atau menerima data product dalam bentuk JSON.
6. Class Master_Principal_*Controller* adalah class yang diturunkan dari CI_*Controller* yang digunakan untuk menyediakan semua fungsionalitas dari halaman Master_Principal_View, Finance_View, Data_Faktur_Pajak_View, Add_Principal_View, dan Edit_Principal_View.
7. Class Principal_Model adalah class yang diturunkan dari CI_Model yang berisi fungsi-fungsi yang digunakan oleh Master_Principal_*Controller* untuk melakukan operasi *database*.
8. prinFactory digunakan oleh Master_Principal_View, Finance_View, Data_Faktur_Pajak_View, Add_Principal_View, dan Edit_Principal_View untuk mengakses fungsi yang terdapat pada Master_Principal_*Controller*.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Rancang bangun aplikasi..., Michael, FTI UMN, 2017



Gambar 3.22 Class Diagram Master Principal

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

E. Pemrograman Sistem

Perancangan dan pembangunan aplikasi berbasis *web* untuk Master Principal dan Master Category menggunakn beberapa perangkat lunak dan perangkat keras. Perangkat keras yang digunakan berupa komputer.

Spesifikasi dari komputer yang digunakan dalam pengerjaan aplikasi berbasis *web* ini adalah sebagai berikut:

1. Processor: Intel(R) Core(TM) i5-3330 CPU @ 3.00GHz
2. RAM: 8 GB
3. Sistem Operasi: Windows 7 64 bit
4. Hard disk: 500 GB

Spesifikasi dari perangkat lunak yang digunakan untuk pembuatan aplikasi berbasis *web* ini adalah sebagai berikut:

1. Notepad++
2. XAMPP
3. SQL Server Driver (sql_srv)

Hasil dari tahapan pemrograman sistem adalah Master Principal dan Master Category. Isi dari kedua aplikasi berbasis *web* tersebut berupa *controller*, model, modul AngularJS dan halaman tampilan yang semuanya berekstensi .php.

Salah satu contoh dari *controller* yang dibuat adalah file "Master_Principal" yang berisi fungsi-fungsi yang digunakan untuk menerima data dalam bentuk JSON lalu melakukan *decode* terhadap data tersebut, mengirim data dalam bentuk JSON ke view dan membentuk tampilan dari suatu halaman.

Gambar 3.23, 3.24, 3.25 merupakan contoh fungsi yang mengembalikan data dalam bentuk JSON, menerima data dalam bentuk JSON, dan membentuk view.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```
public function get_all_product_list($no_prin){
  $query = $this->Principal->get_product_list($no_prin)->result_array();
  $this->output->set_content_type('application/json')->set_output(json_encode($query));
}
```

Gambar 3.23 Fungsi Controller Master Category yang Mengembalikan

Data dalam Bentuk JSON

```
public function update_fp(){
  $postdata = file_get_contents("php://input");
  $data = json_decode($postdata);
  $this->Principal->update_fp($data->no_prin,$data->group,$data->CB_ID,$data->BI_ID,$data->VA_CODE);
  echo $result = '{"status":"success"}';
}
public function add_newFp(){
  $postdata = file_get_contents("php://input");
  $data = json_decode($postdata);
  $this->Principal->add_newFP_data($data->no_prin,$data->group,$data->CB_ID,$data->BI_ID,$data->VA_CODE);
  echo $result = '{"status":"success"}';
}
```

Gambar 3.24 Fungsi Controller Master Principal yang Menerima

Data dengan Format JSON

```
public function generateTemplate(){
  $data['style'] = $this->load->view('include/style', NULL, TRUE);
  $data['script'] = $this->load->view('include/script', NULL, TRUE);
  $data['datatables_script'] = $this->load->view('include/datatables_script', NULL, TRUE);
  $data['datatables_style'] = $this->load->view('include/datatables_style', NULL, TRUE);
  $data['navbar'] = $this->load->view('template/navbar', NULL, TRUE);
  $data['angularJS'] = $this->load->view('include/angularJS', NULL, TRUE);
  return $data;
}
public function index($group,$prev_kode = 0){
  $data = $this->generateTemplate();
  $data['group']     = $group;
  $data['prev_kode'] = $prev_kode;
  $this->load->view('Master_Principal',$data);
}
```

Gambar 3.25 Fungsi Controller Master Principal untuk Membentuk View

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Untuk melakukan operasi *database*, *controller* akan memanggil Model. Fungsi pada model akan memberikan hasil *query* ke *controller* untuk dikonversi dalam format JSON.

Gambar 3.26 contoh fungsi pada Category_Model yang melakukan *query* ke *database* dan mengembalikan hasil *query*nya.

```
public function getSubCategory($Cat_ID)
{
  $query = $this->db->query("dbo.spCenMst_Category_View 6, '" . trim($Cat_ID) . "', '");
  return $query;
}
public function getCategoryByName($name){
  $query = $this->db->query("century_master.dbo.spCenMst_Category_View 11, '', '" . str_replace("'", "`",$name) . "'");
  return $query;
}
public function getCategoryByKode($code){
  $query = $this->db->query("century_master.dbo.spCenMst_Category_View 10, '" . str_replace("'", "`",$code) . "', '");
  return $query;
}
```

Gambar 3.26 Fungsi pada Category_Model

Controller yang telah dibuat dapat dipanggil dari view secara asinkronus menggunakan modul AngularJS yang di dalamnya terdapat Factory. Factory adalah fungsi yang mengembalikan objek-objek.

Gambar 3.27 contoh prinFactory yang terdapat pada modul principal.factory yang sudah diberikan pengaturan untuk melakukan konversi data yang dikirim ke *controller* dalam bentuk JSON.

```
angular.module('principal.factory',[]).config(function($httpProvider){
  $httpProvider.defaults.transformRequest = function(data,getHeaders){
    var headers = getHeaders();
    return JSON.stringify(data);
  }
}).factory('prinFactory',function($http){
  return{
    edit_principal:function(group,principalObj){
      return $http.post(base_url+'index.php/Master_Principal/edit_principal'+'/'+group,principalObj);
    },
    add_mass_cp:function(no_prin,obj){
      return $http.post(base_url+'index.php/Master_Principal/add_mass_cp' + '/' + no_prin,obj);
    },
```

Gambar 3.27 Pengaturan dan Fungsi-Fungsi pada Factory

Rancang bangun aplikasi..., Michael, FTI UMN, 2017

Gambar 3.28 contoh pemanggilan fungsi-fungsi pada prinFactory pada halaman Master_Principal_View

```
$scope.get_principal_info = function(){
   prinFactory.get_PrincipalData($scope.principal.id).then(function(response){
    $scope.principal.name          = response.data[0].PRI_NAME;
    $scope.principal.address       = response.data[0].PRI_ADDRESS;
    $scope.principal.city_code     = response.data[0].PRI_CITYCODE;
    $scope.principal.kec_id        = response.data[0].PRI_KECID;
    $scope.principal.postal_code   = response.data[0].PRI_POSTALCODE;
    $scope.principal.phone_code    = response.data[0].PRI_PHONECODE;
    $scope.principal.phone         = response.data[0].PRI_TELPNUMBER;
    $scope.principal.fax           = response.data[0].PRI_FAXNUMBER;
    $scope.principal.contact       = response.data[0].PRI_CONTACT;
    $scope.principal.activeYN      = response.data[0].PRI_ACTIVEYN;
    $scope.principal.EDLP          = response.data[0].PRI_EDLP;
    $scope.principal.city_name     = (response.data[0].KOTA_NAME).trim();
    $scope.principal.kec_name      = (response.data[0].KEC_NAMA).trim();
    $scope.principal.dpPercentage = response.data[0].PRI_DPPERCENTAGE;
    $scope.principal.dpValueRupiah  = response.data[0].PRI_DPVALUERUPIAH;
    if(response.data[0].PRI_DPPERIODSTART != null ){
      $scope.principal.dpPeriodStart  = response.data[0].PRI_DPPERIODSTART.replace(' ','T');
    }
    else{
      $scope.principal.dpPeriodStart  = response.data[0].PRI_DPPERIODSTART;
    }
    if(response.data[0].PRI_DPPERIODEND != null ){
      $scope.principal.dpPeriodEnd = response.data[0].PRI_DPPERIODEND.replace(' ','T');
    }
    else{
      $scope.principal.dpPeriodEnd  = response.data[0].PRI_DPPERIODEND;
    }
    $scope.principal.status     = response.data[0].PRI_ACTIVEYN;
    return prinFactory.get_ContactPersonData($scope.principal.id);
  }).then(function(response){
    $scope.contactPerson = response.data;
  });
}
```

Gambar 3.28 Pemanggilan Fungsi yang Terdapat pada prinFactory

Setelah Javascript telah berhasil mendapatkan data dalam bentuk JSON maka halaman Master_Principal_View akan menampilkan data yang diperoleh tersebut.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.29 contoh pengaksesan objek principal yang menampung data yang dikirim oleh *controller*.

```
<table datatable="ng" class="table" cellspacing="0" width="100%">
  <thead>
    <tr>
      <th>NAMA</th>
      <th>JABATAN</th>
      <th>JENIS</th>
      <th>KD AREA</th>
      <th>NOMOR</th>
      <th>Urut</th>
      <th>PIC JP</th>
    </tr>
  </thead>
  <tbody >
    <tr ng-repeat="x in contactPerson">
      <td>{{x.Nama}}</td>
      <td>{{x.Jabatan}}</td>
      <td>{{x.Jenis}}</td>
      <td>{{x.KodeArea}}</td>
      <td>{{x.Nomor}}</td>
      <td>{{x.Urut}}</td>
      <td>{{x.PIC_JP}}</td>
    </tr>
  </tbody>
</table>
```

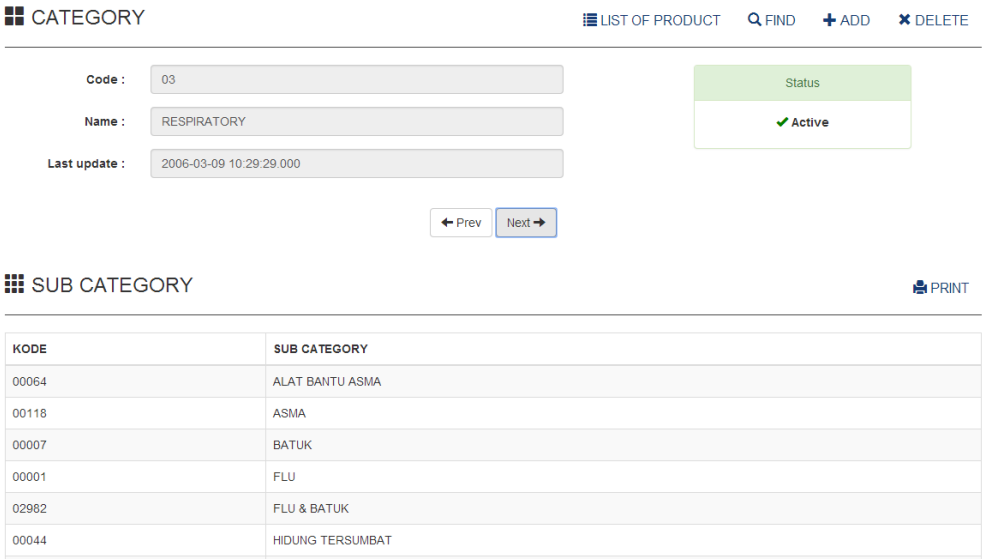
Gambar 3.29 Mengakses Objek contactPerson untuk Ditampilkan pada Tabel

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

F. Implementasi Sistem

Pengujian sistem atau modul yang telah dikembangkan akan dilakukan dengan menunjukan dan menjelaskan gambar (*screenshoot*) dari aplikasi yang menunjukan fungsionalitas dan tampilan dari kedua aplikasi yang dikembangkan.

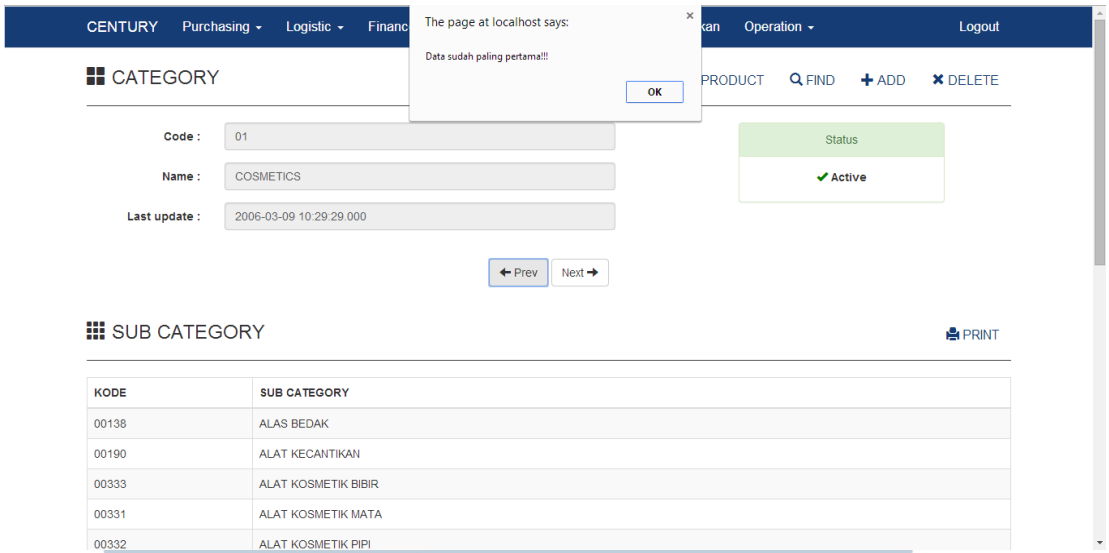
Gambar 3.30 merupakan halaman dari Master_Category_View yang merupakan halaman utama.



Gambar 3.30 Master_Category_View

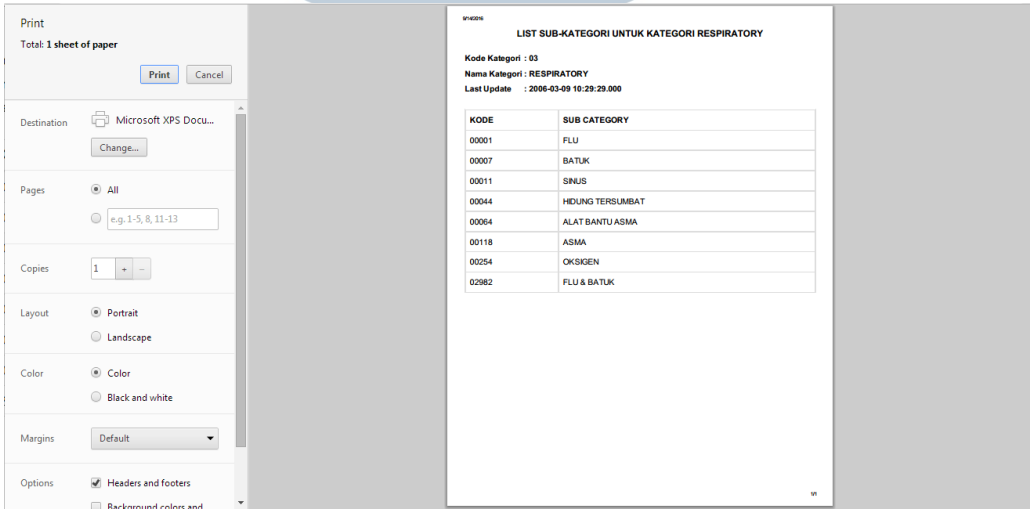
User bisa melakukan mencari kategori secara satu persatu melalui tombol next dan prev yang telah disediakan, apabila data sudah paling pertama dan user menekan tombol next maka akan muncul notifikasi, hal yang sama juga terjadi ketika user menekan tombol next pada data yang paling terakhir.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.31 Notifikasi Data Sudah Paling Pertama

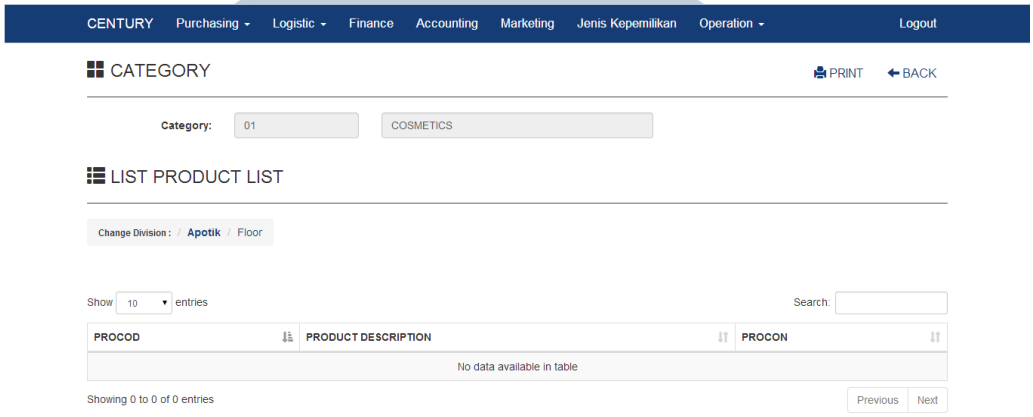
Apabila user ingin mencetak daftar subkategori, user hanya perlu menekan tombol print yang terdapat di pojok kanan tabel sub kategori, berikut adalah tampilan dari halaman print preview.



Gambar 3.32 Halaman Print Preview

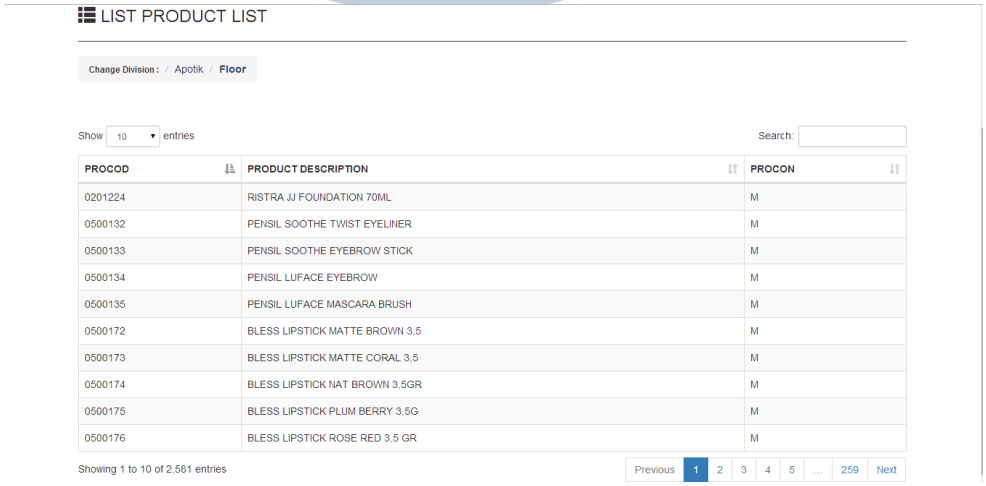
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Ketika user melalukan navigasi ke halaman List_Product_View maka tampilan awal adalah daftar produk untuk divisi Apotik.



Gambar 3.33 Daftar Produk Kategori Kosmetik Divisi Apotik

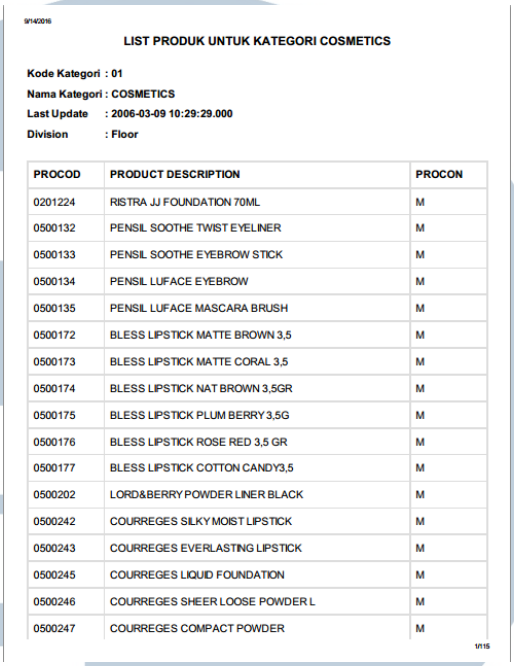
User bisa mengubah divisi melalui tombol navigasi telah tersedia, Gambar 3.34 merupakan contoh daftar produk untuk divisi Floor.



Gambar 3.34 Daftar Produk Divisi Floor

User bisa mencetak daftar produk menggunakan tombol print yang terdapat pada menu navigasi Gambar 3.35 merupakan contoh dari halaman print preview untuk mencetak daftar produk.

UNIVERSITAS MULTIMEDIA NUSANTARA



9/14/2016

LIST PRODUK UNTUK KATEGORI COSMETICS

Kode Kategori : 01
Nama Kategori : COSMETICS
Last Update : 2006-03-09 10:29:29.000
Division : Floor

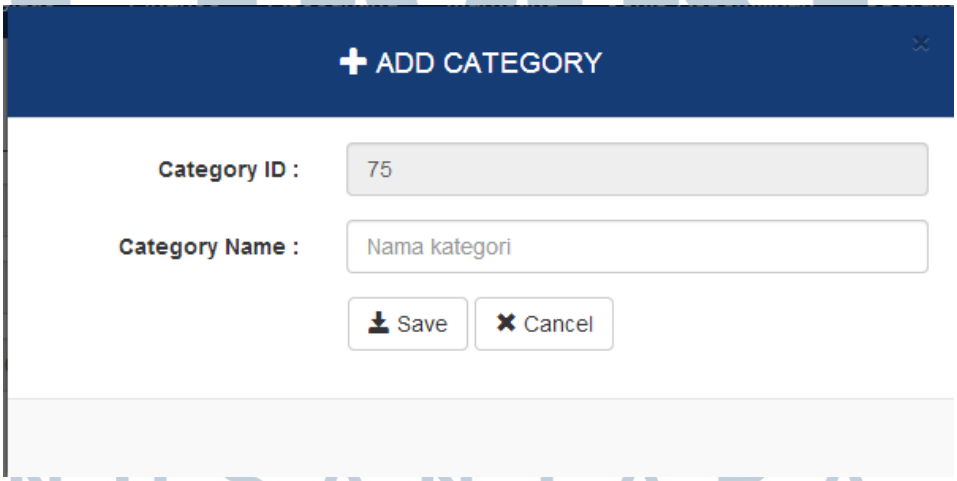
PROCOD	PRODUCT DESCRIPTION	PROCON
0201224	RISTRA JJ FOUNDATION 70ML	M
0500132	PENSIL SOOTHE TWIST EYELINER	M
0500133	PENSIL SOOTHE EYEBROW STICK	M
0500134	PENSIL LUFACE EYEBROW	M
0500135	PENSIL LUFACE MASCARA BRUSH	M
0500172	BLESS LIPSTICK MATTE BROWN 3,5	M
0500173	BLESS LIPSTICK MATTE CORAL 3,5	M
0500174	BLESS LIPSTICK NAT BROWN 3,5GR	M
0500175	BLESS LIPSTICK PLUM BERRY 3,5G	M
0500176	BLESS LIPSTICK ROSE RED 3,5 GR	M
0500177	BLESS LIPSTICK COTTON CANDY3,5	M
0500202	LORD&BERRY POWDER LINER BLACK	M
0500242	COURREGES SILKY MOIST LIPSTICK	M
0500243	COURREGES EVERLASTING LIPSTICK	M
0500245	COURREGES LIQUID FOUNDATION	M
0500246	COURREGES SHEER LOOSE POWDER L	M
0500247	COURREGES COMPACT POWDER	M

1/115

Gambar 3.35 Print Preview Daftar Produk

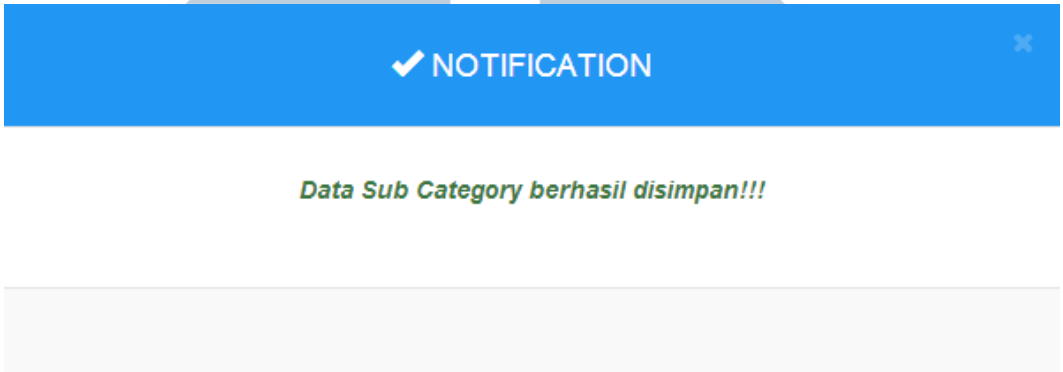
Pada halaman utama Master_Category_View user bisa menambah kategori baru dan mengubah status kategori menjadi tidak aktif.

Gambar 3.36 merupakan gambar kotak dialog yang digunakan untuk memasukan kategori baru.



Gambar 3.36 Kotak Dialog Add Category

Ketika sudah selesai menambah kategori maka akan muncul dialog notifikasi yang ditunjukan gambar 3.37



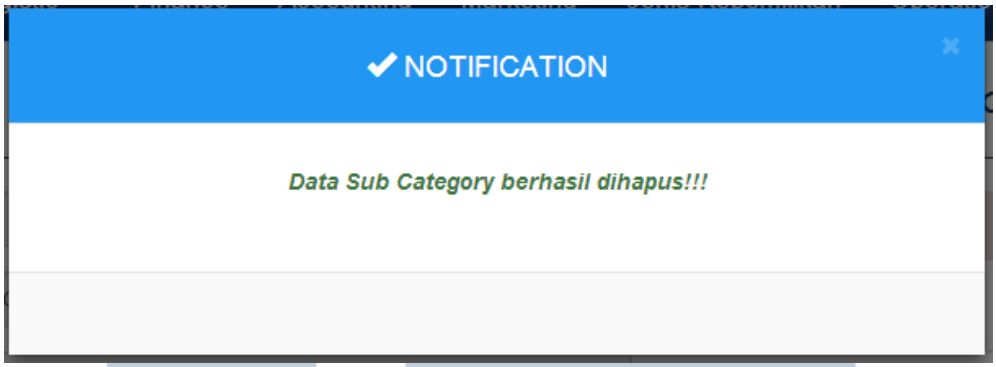
Gambar 3.37 Kotak Dialog Notifikasi Data Berhasil Disimpan

Gambar 3.38 merupakan gambar kotak dialog yang digunakan untuk melakukan konfirmasi untuk mengubah status kategori menjadi tidak aktif.

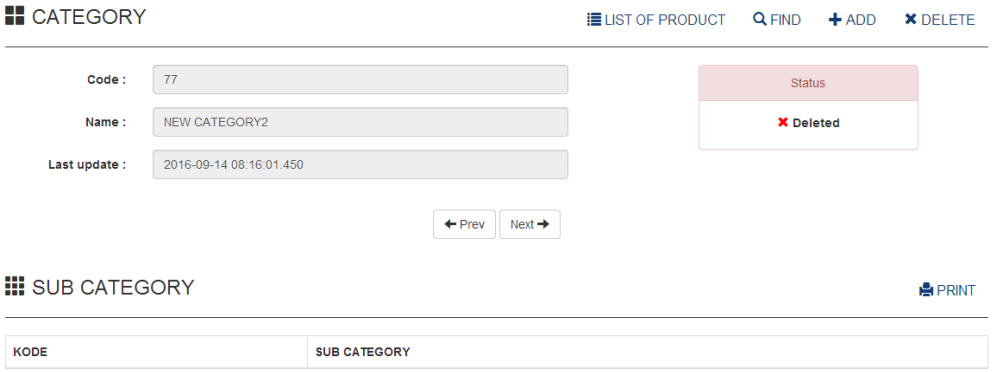
Gambar 3.38 Kotak Dialog Konfirmasi Hapus

Jika konfirmasi telah dilakukan akan muncul kotak dialog yang memberitahu user bahwa data sudah berhasil dihapus / statusnya menjadi tidak aktif.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

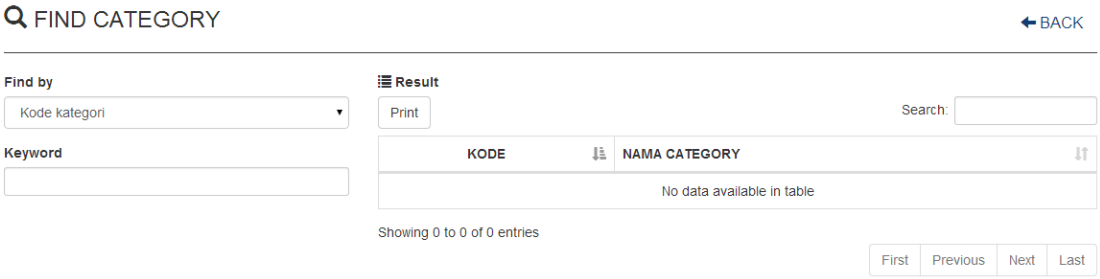


Gambar 3.39 Notifikasi Kategori Berhasil Dihapus



Gambar 3.40 Status Kategori Menjadi Deleted

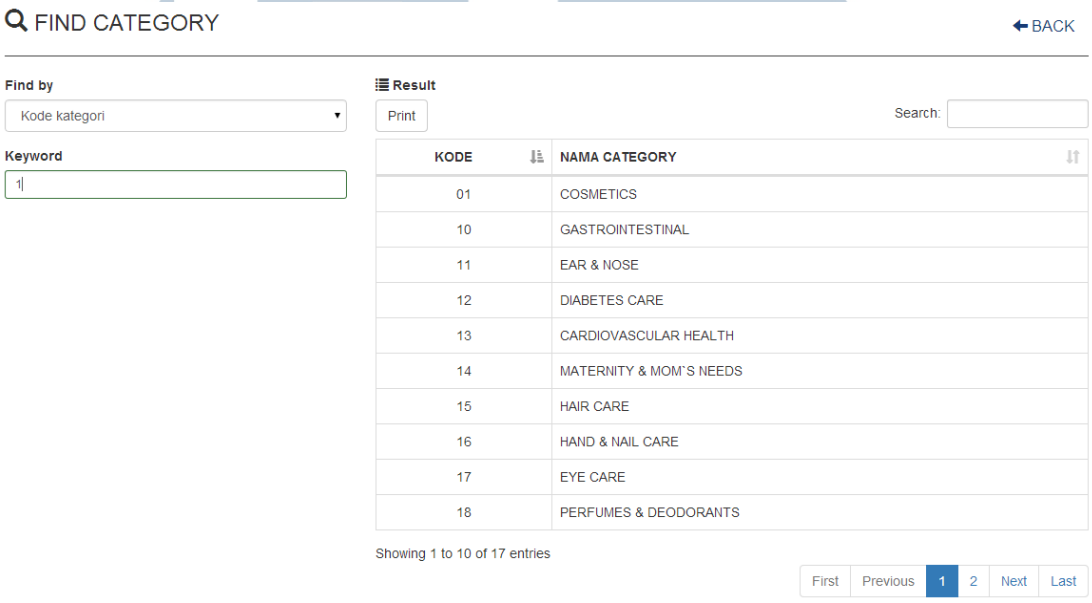
Jika user mengakses halaman Find_Category_View maka tampilan awalnya adalah seperti Gambar 3.41



Gambar 3.41 Tampilan Awal Find_Category_View

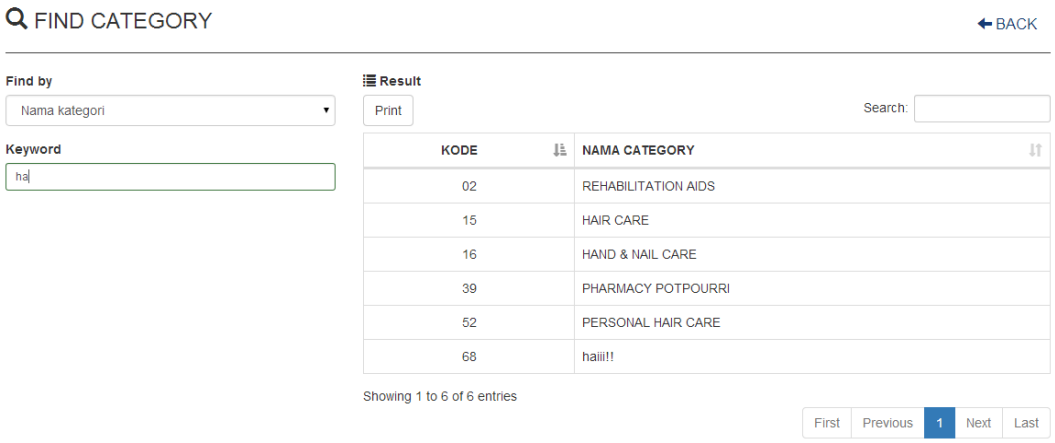
MULTIMEDIA NUSANTARA

User bisa melakukan pencarian berdasarkan kode kategori atau menggunakan nama kategori. Gambar 3.42 menunjukan pencarian kategori oleh user menggunakan kata kunci berupa kode kategori.



Gambar 3.42 Pencarian Kategori Menggunakan Kode

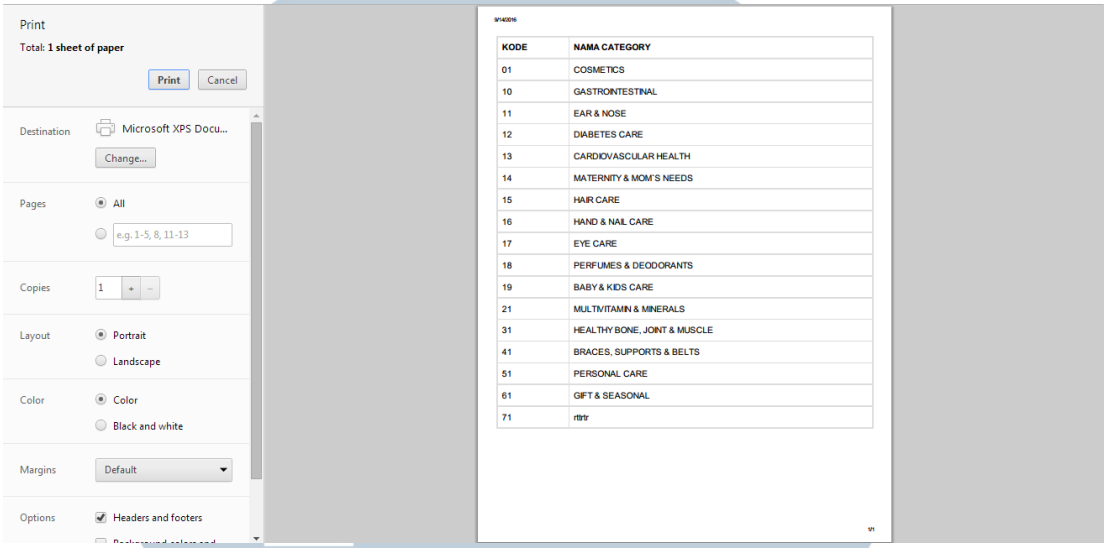
Gambar 3.43 menunjukan pencarian kategori oleh user menggunakan kata kunci berupa nama kategori.



Gambar 3.43 Melakukan Pencarian Kategori Berdasarkan Nama

NUSANTARA

Apabila user menekan tombol print maka ditampilkan halam print preview seperti berikut pada tampak Gambar 3.44

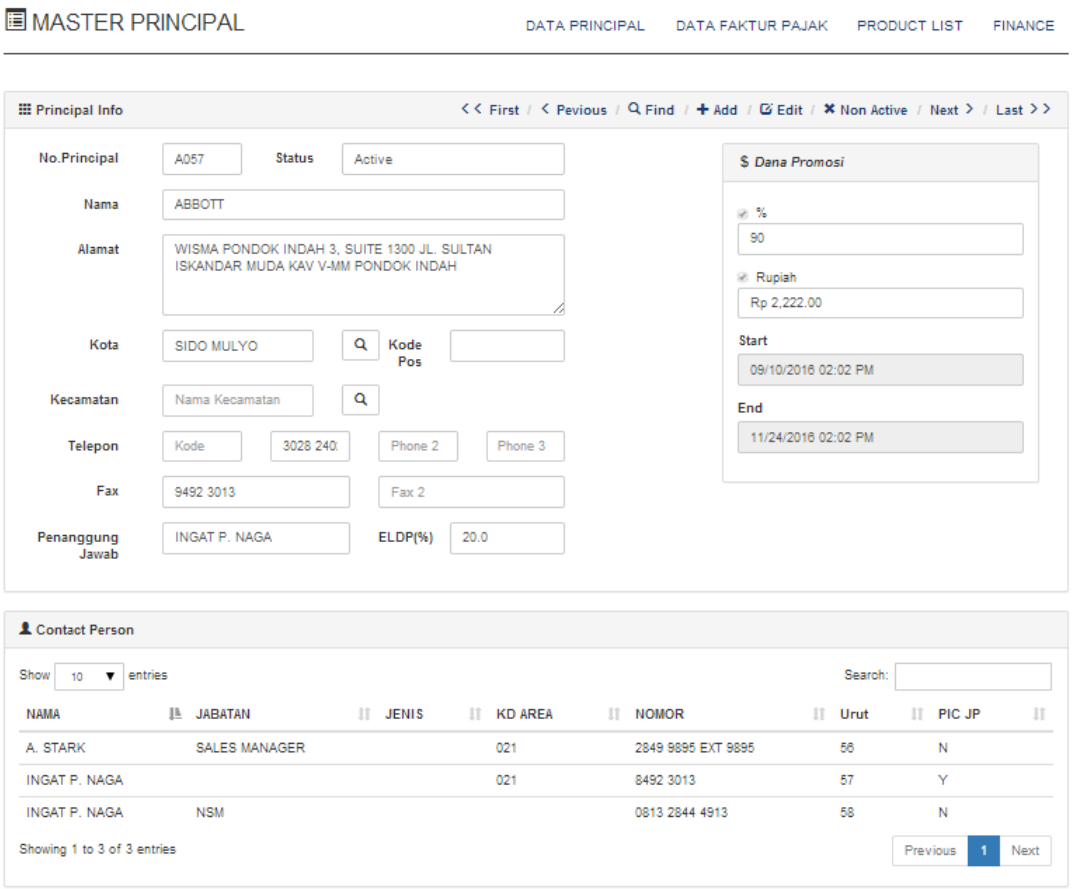


Gambar 3.44 Halaman Print Preview Find Category

Dengan demikian aplikasi berbasis *web* untuk Master Category sudah selesai diuji, berikutnya adalah pengujian pada Master Principal.

Gambar 3.45 adalah halaman utama untuk Master Principal dimana menampilkan data principal paling pertama.

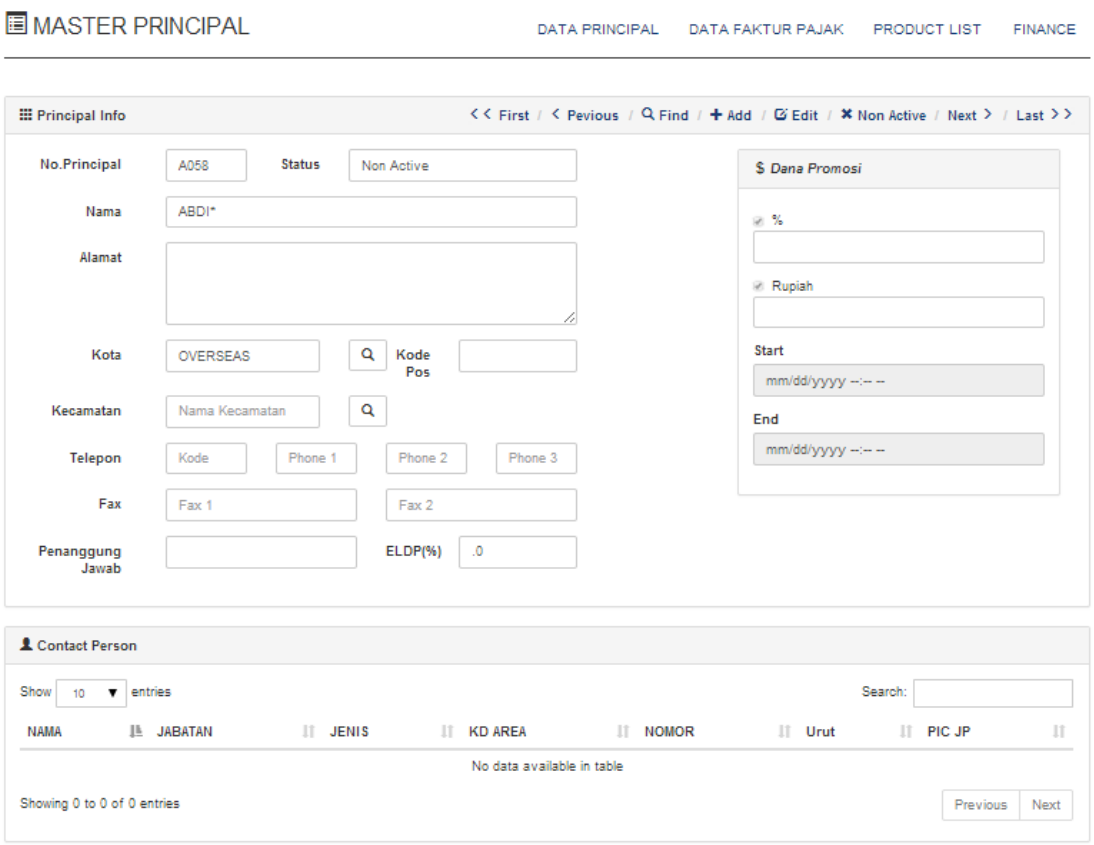
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.45 Halaman Utama Master Principal

User bisa melakukan navigasi untuk menampilkan data principal selanjutnya atau sebelumnya melalui tombol next dan prev yang telah disediakan.

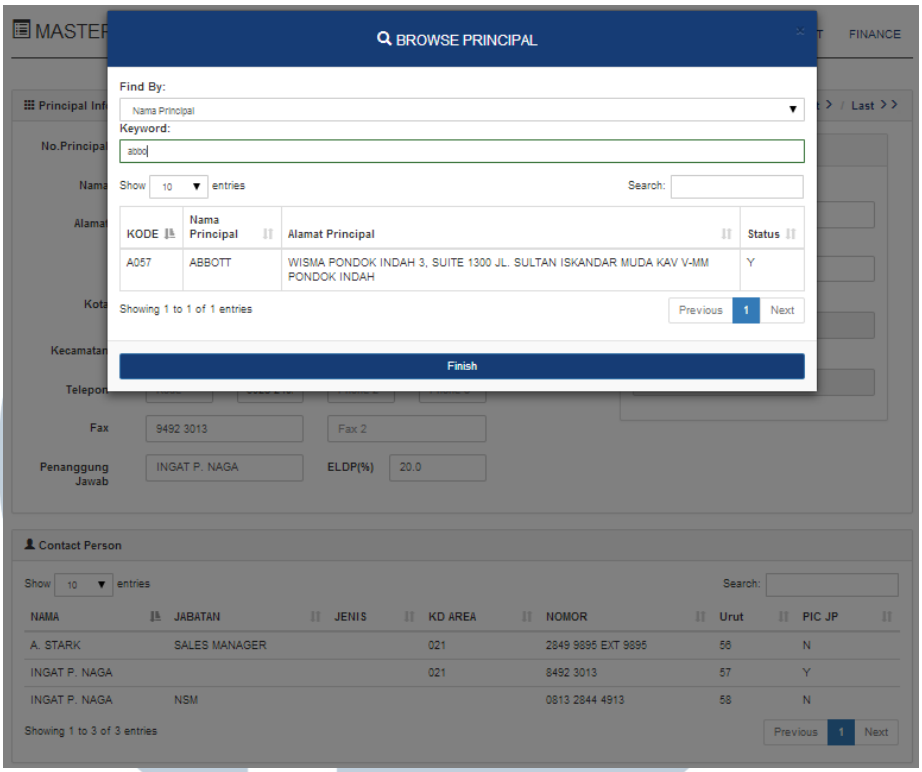
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.46 Menekan Tombol Next

User bisa melakukan navigasi secara langsung untuk menemukan principal yang spesifik apabila menekan tombol Find yang ada pada menu navigasi sehingga kotak dialog pencarian akan ditampilkan.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

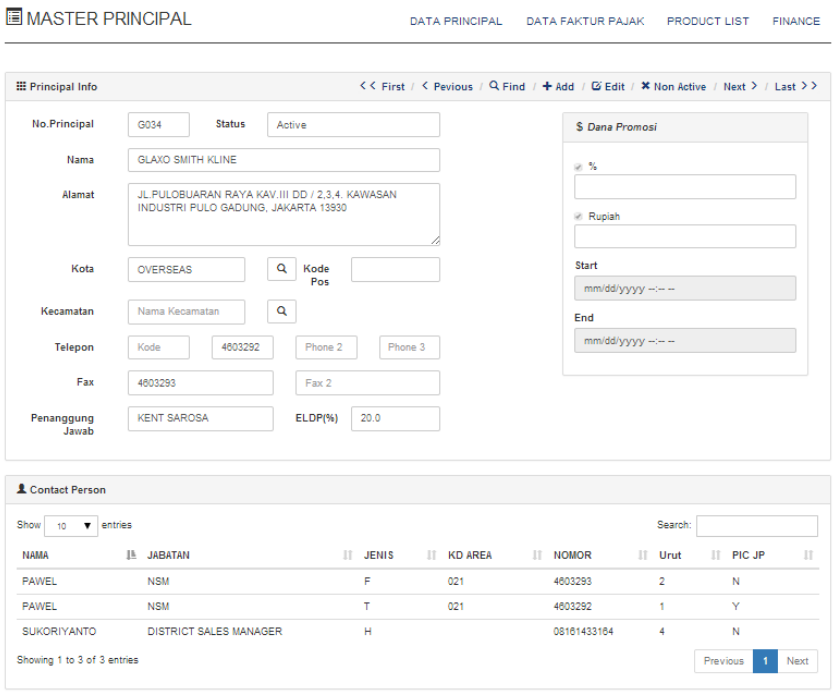


Gambar 3.47 Kotak Dialog Pencarian Principal

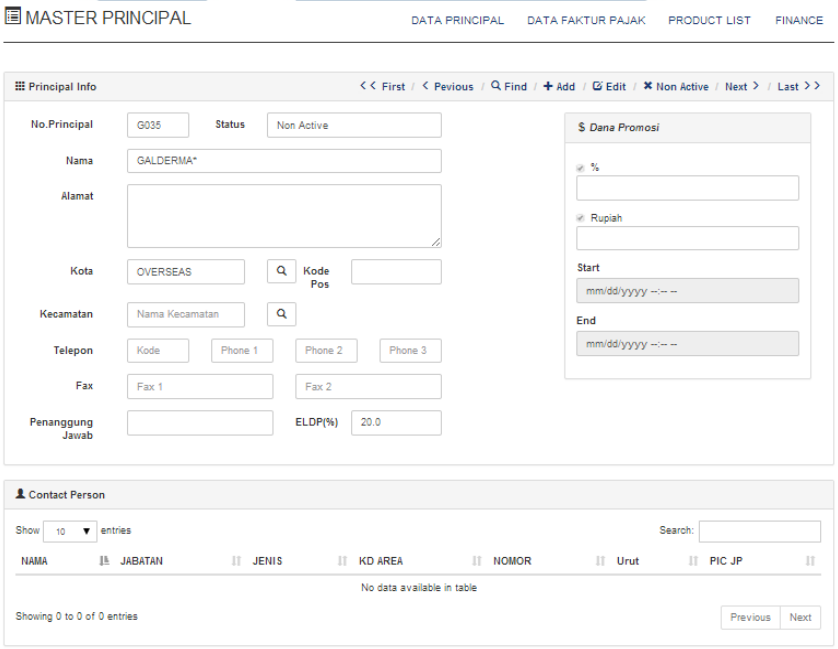
Dalam melakukan pencarian user bisa memilih metode untuk melakukan pencarian berdasarkan nama principal atau kode principal apabila hasil pencarian sudah ditampilkan maka user bisa menekan hasil pencarian tersebut untuk ditampilkan pada halaman utama.

Apabila user menekan tombol Non Active maka status principal yang tadinya aktif menjadi tidak aktif.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



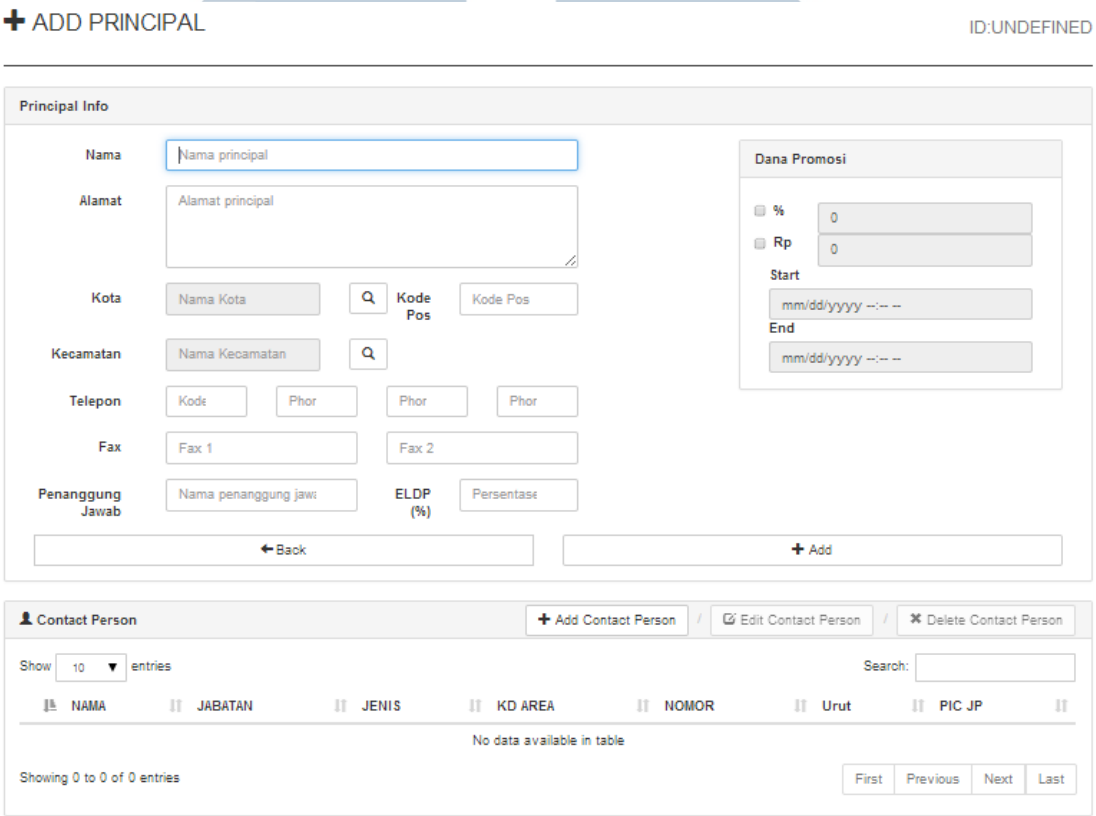
Gambar 3.48 Status Principal Masih Aktif



Gambar 3.49 Status Principal Menjadi Tidak Aktif

UNIVERSITAS
MULTIMEDIA
NUSANTARA

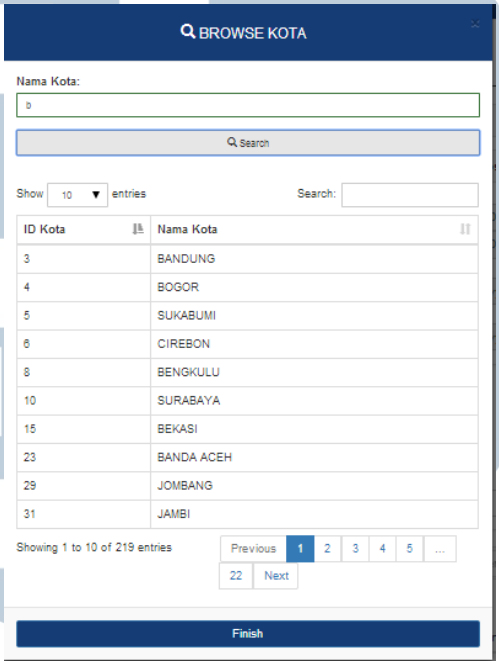
Apabila user menekan tombol Add maka halaman Add Principal akan ditampilkan, tampilan awal halaman tersebut adalah seperti berikut



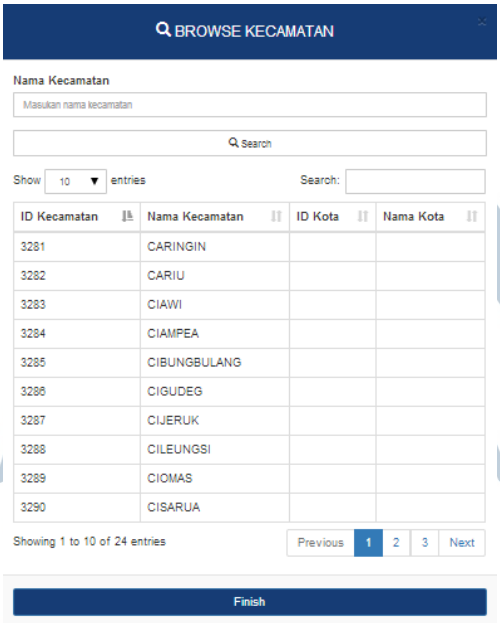
Gambar 3.50 Halaman Add Principal

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Untuk memberikan input pada kecamatan dan kota user harus menekan tombol yang telah disediakan sehingga kotak dialog kota dan kecamatan akan ditampilkan tampak seperti gambar berikut

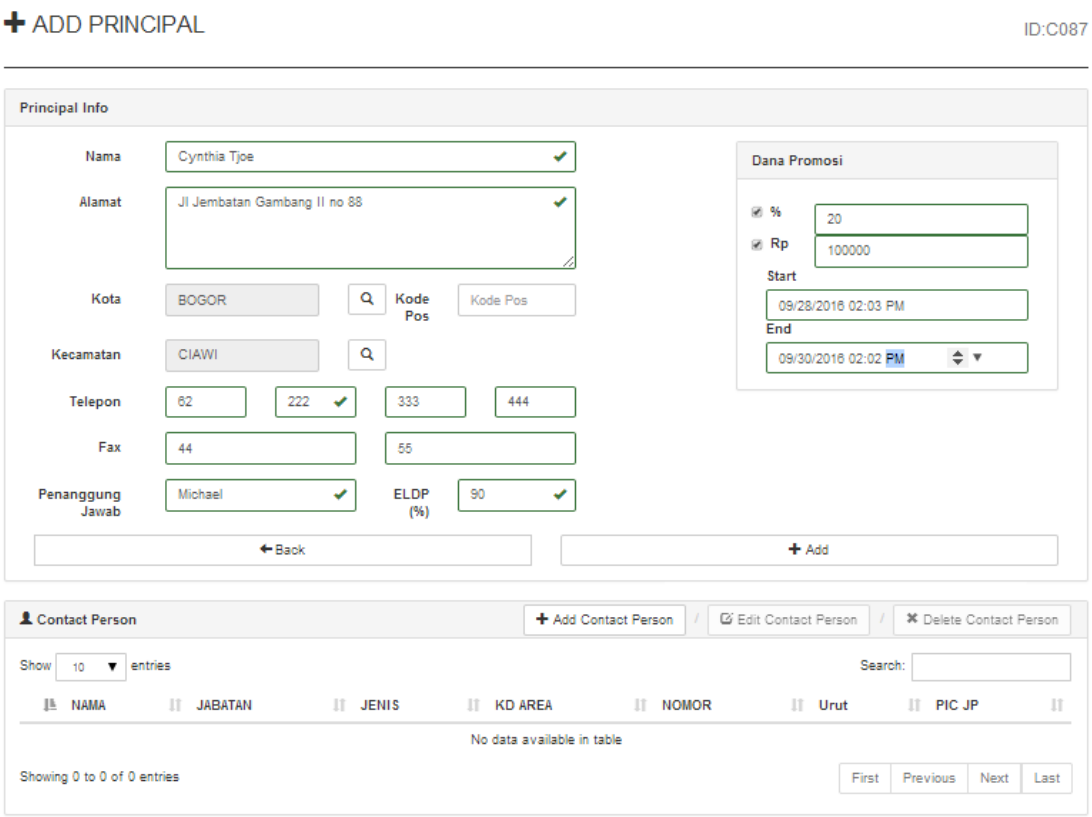


Gambar 3.51 Kotak Dialog Browse Kota



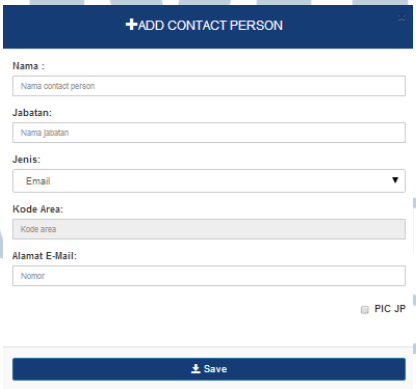
Gambar 3.52 Kotak Dialog Browse Kecamatan

Form akan melakukan validasi secara langsung tiap kali user memberikan masukan pada form yang ditunjukan dengan gambar berikut



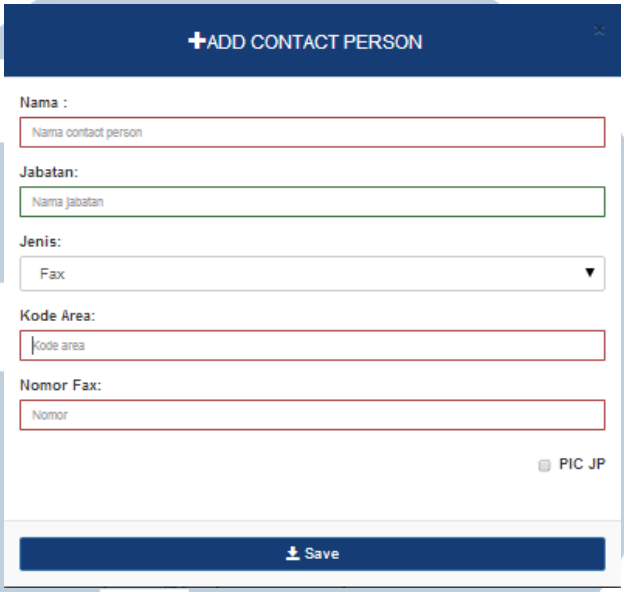
Gambar 3.53 Validasi Form Secara Langsung

Untuk menambah contact person, user harus menekan tombol Add Contact Person pada tabel daftar contact person sehingga kotak dialog Add Contact person akan ditampilkan.



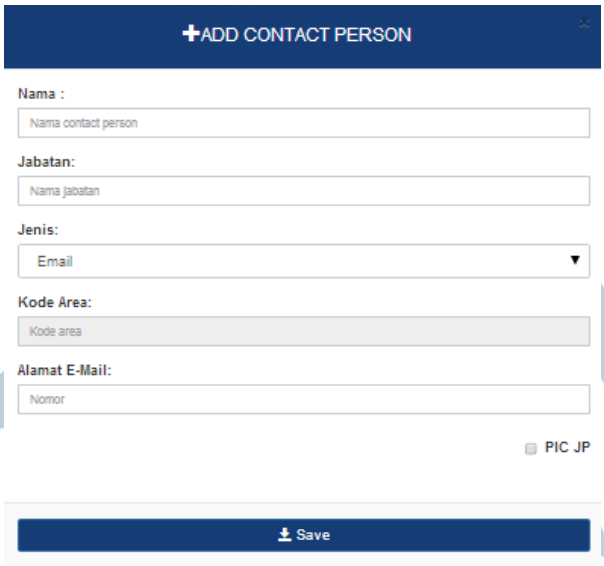
Gambar 3.54 Kotak Dialog Add Contact Person

Validasi secara langsung akan dilakukan dengan memberikan warna field pada formulir bewarna merah apabila terjadi kesalahan.



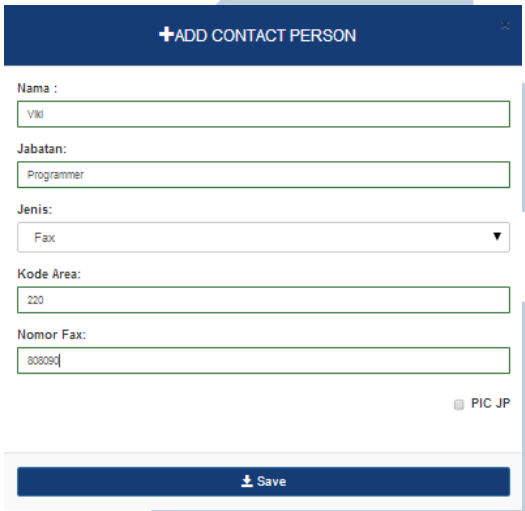
Gambar 3.55 Validasi Form Contact Person

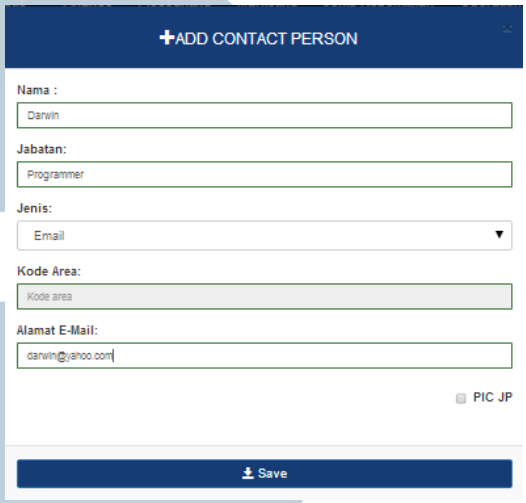
Dropdown untuk jenis contact terdiri dari fax, email,telepon, dan telpon genggam. Field kode area akan disabled jika jenis dari contact person adalah email

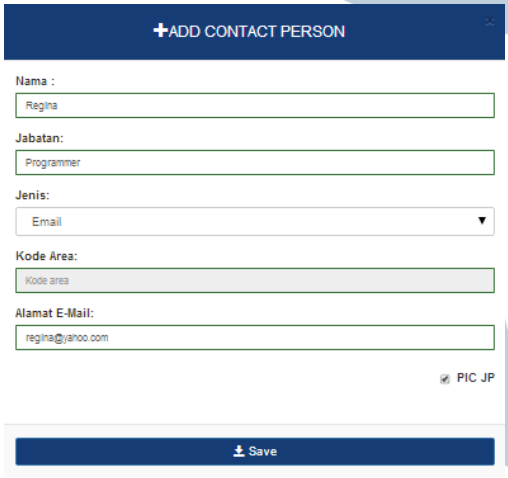


Gambar 3.56 Disable Kode Area Jika Jenis Contact Email

Gambar berikut merupakan pengujian untuk menambahkan 3 buah contact person ke daftar tabel contact person

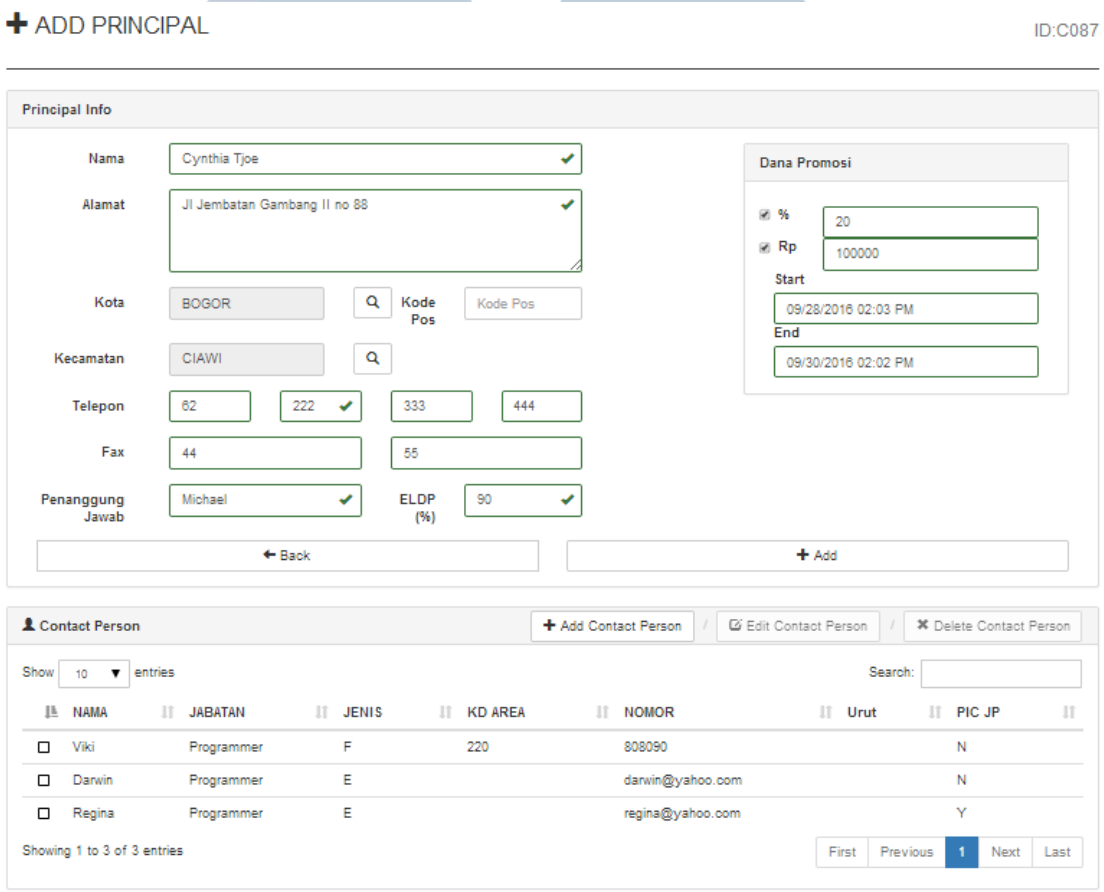






UNIVERSITAS
MULTIMEDIA
NUSANTARA

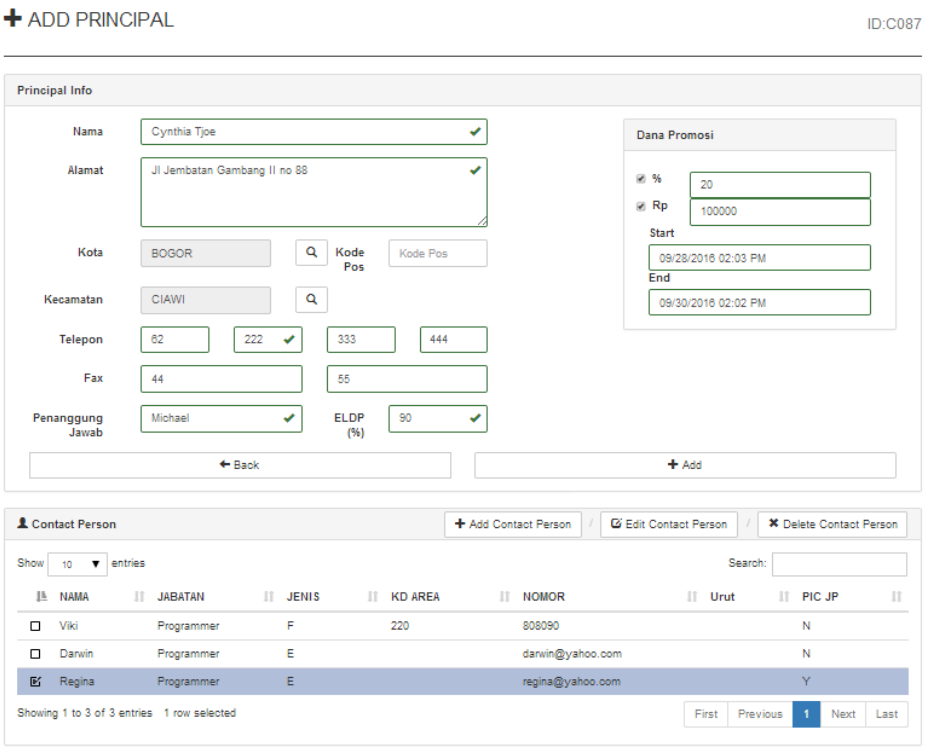
Setelah selesai maka contact person tersebut terdapat pada daftar contact person di halaman Add_Principal_View.



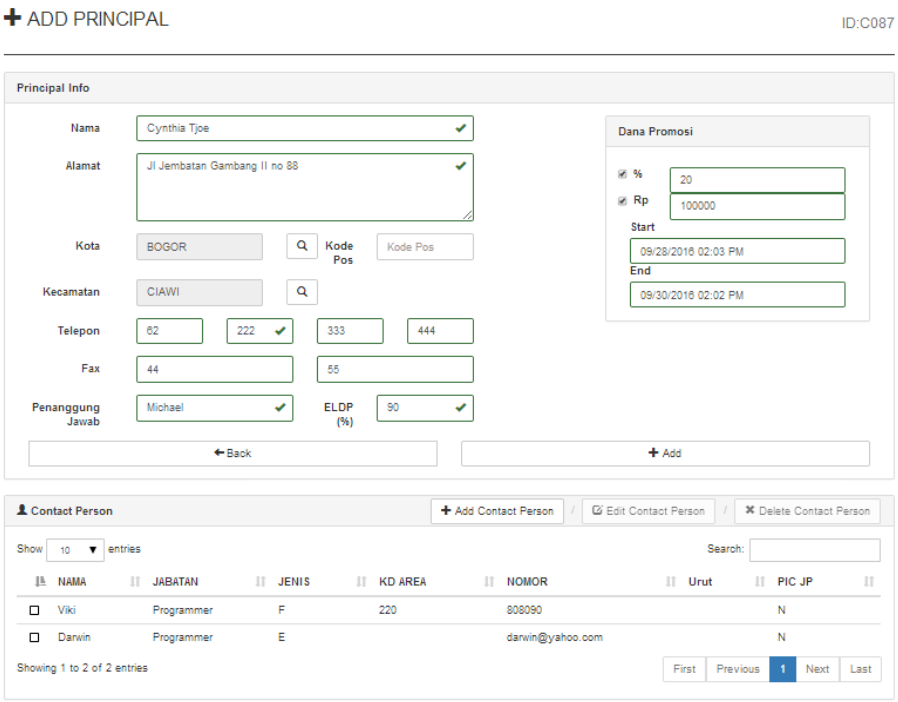
Gambar 3.57 Penambahan Contact Person

User juga bisa menghapus atau mengubah data contact person yang dipilih pada daftar contact person yang sudah dibuat dengan memilih contact person yang akan dihapus atau diubah kemudian menekan tombol Delete Contact Person atau Edit Contact Person.

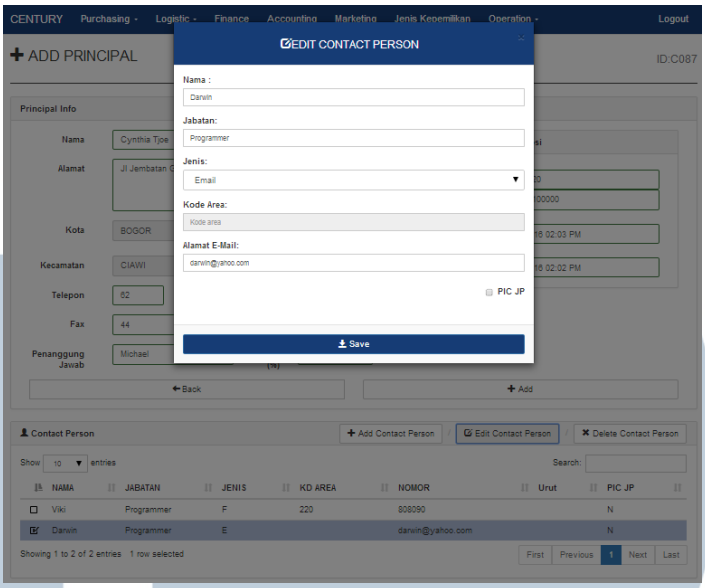
UNIVERSITAS MULTIMEDIA NUSANTARA



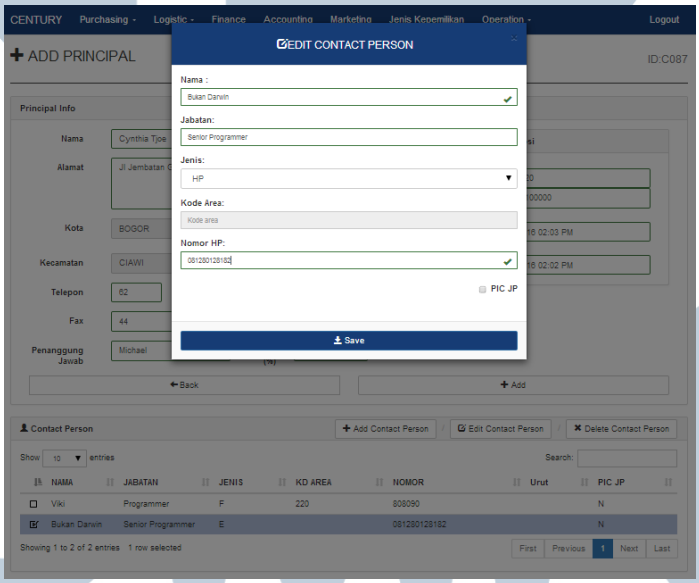
Gambar 3.58 Memilih Contact Person yang Akan Dihapus



Gambar 3.59 Contact Person Berhasil Dihapus

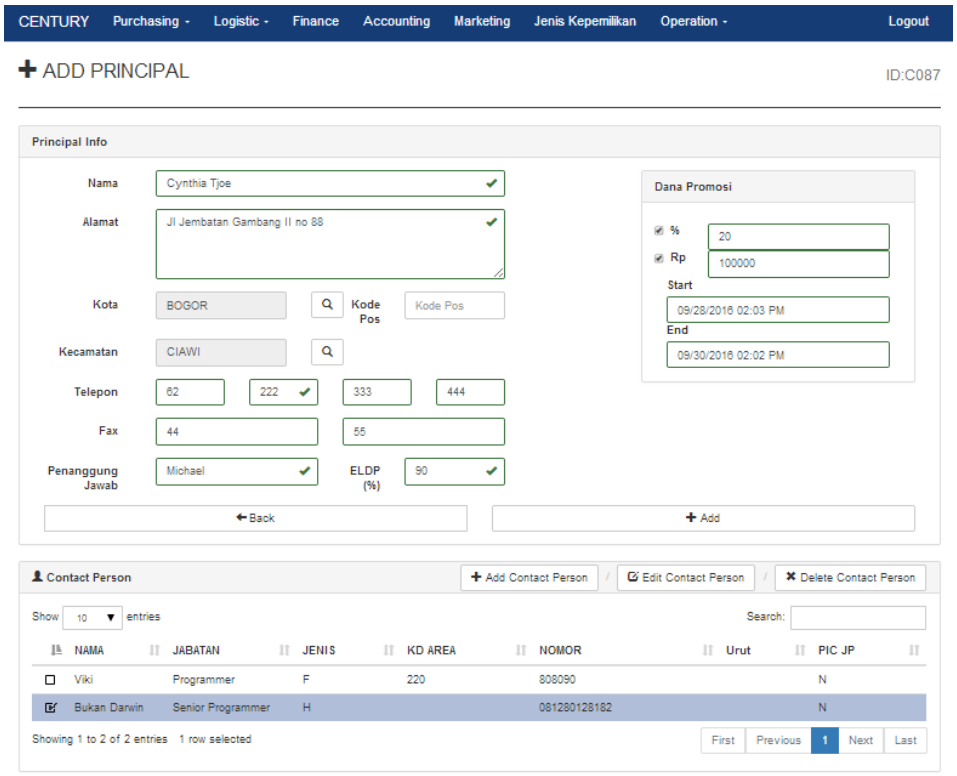


Gambar 3.60 Memilih Contact Person yang Akan Diubah



Gambar 3.61 Melakukan Pengubahan Contact Person

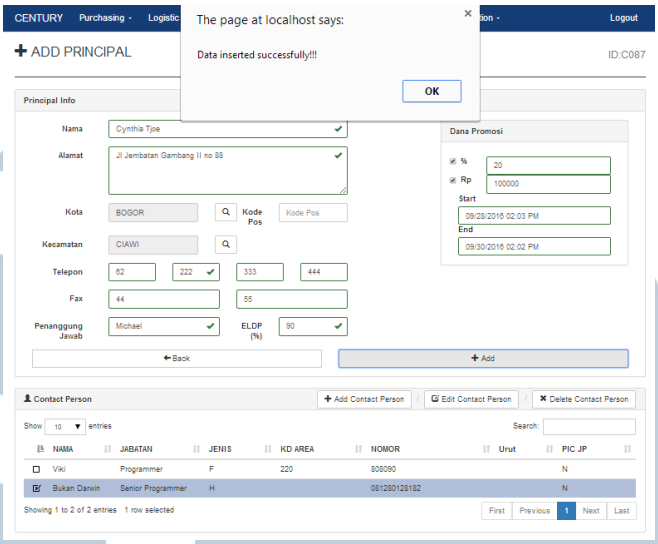
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.62 Data Contact Person yang Telah Berhasil Diubah

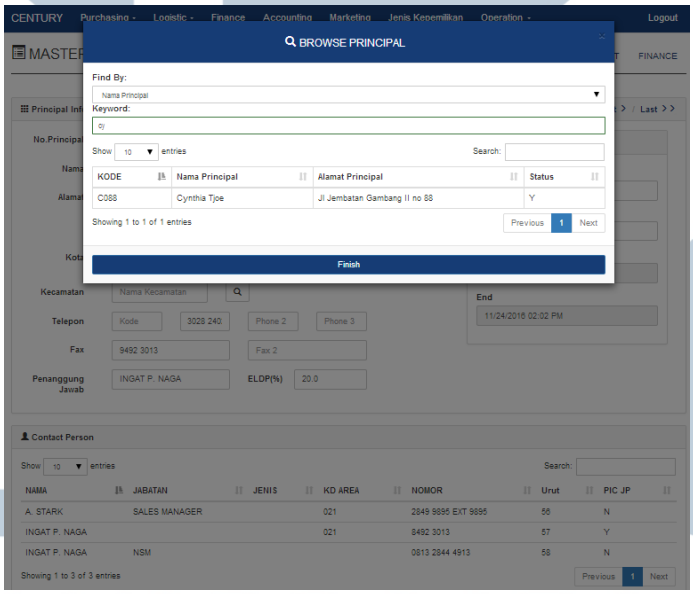
Setelah selesai memasukan data principal disertai dengan daftar contact personnya, user dapat melakukan submit untuk menambahkan data principal yang baru ke *database*. Jika berhasil maka akan muncul kotak dialog notifikasi seperti berikut

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



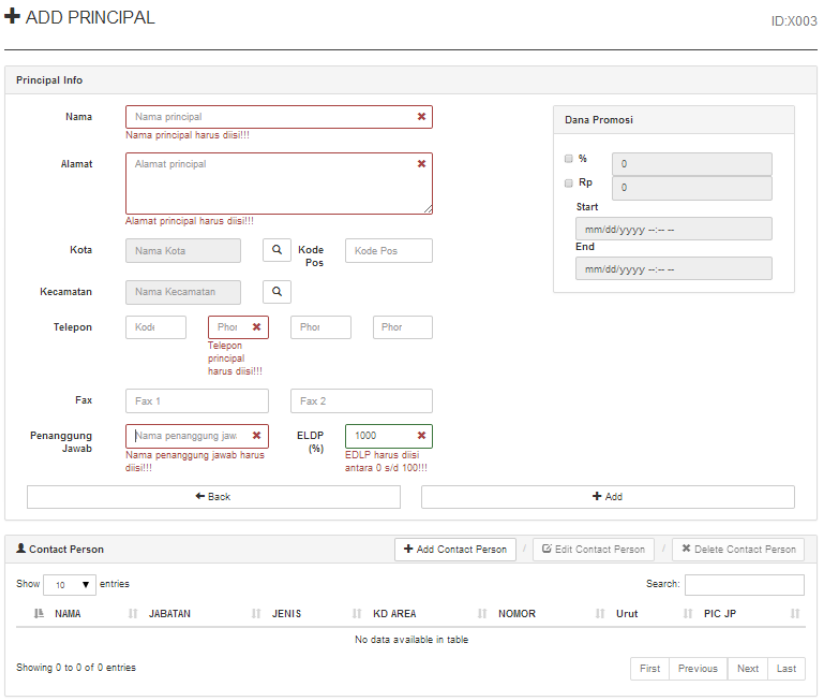
Gambar 3.63 Notifikasi Data Principal Baru Tersimpan

Untuk mengujinya lagi dilakukan pencarian pada data baru yang disimpan ditunjukan pada Gambar 3.64 terlihat bahwa hasil pencarian terhadap data baru berhasil dilakukan.



Formulir Add_Principal akan menampilkan tampilan error seperti berikut apabila ada masukan yang tidak sesuai format yang telah ditentukan.

UNIVERSITAS MULTIMEDIA NUSANTARA

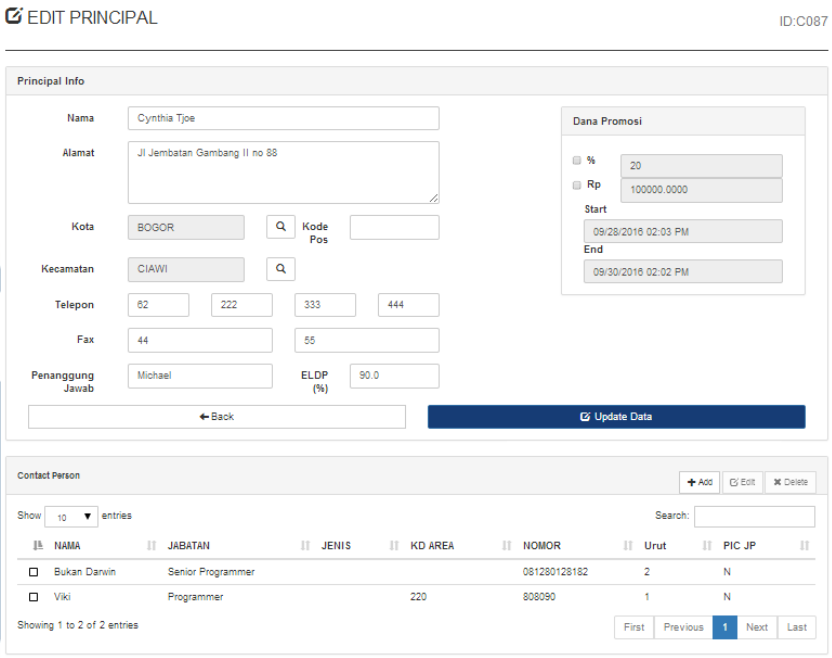


Gambar 3.64 Kesalahan Pada Form Add Principal

Halaman Edit_Principal_View mempunyai cara kerja yang sama dengan Add Principal yang berbeda adalah pada halaman Edit_Principl_View formulir menampilkan data principal yang akan diubah.

Gambar 3.65 menunjukan halaman Edit_Principal_View ketika halaman tersebut ditampilkan pertama kali.

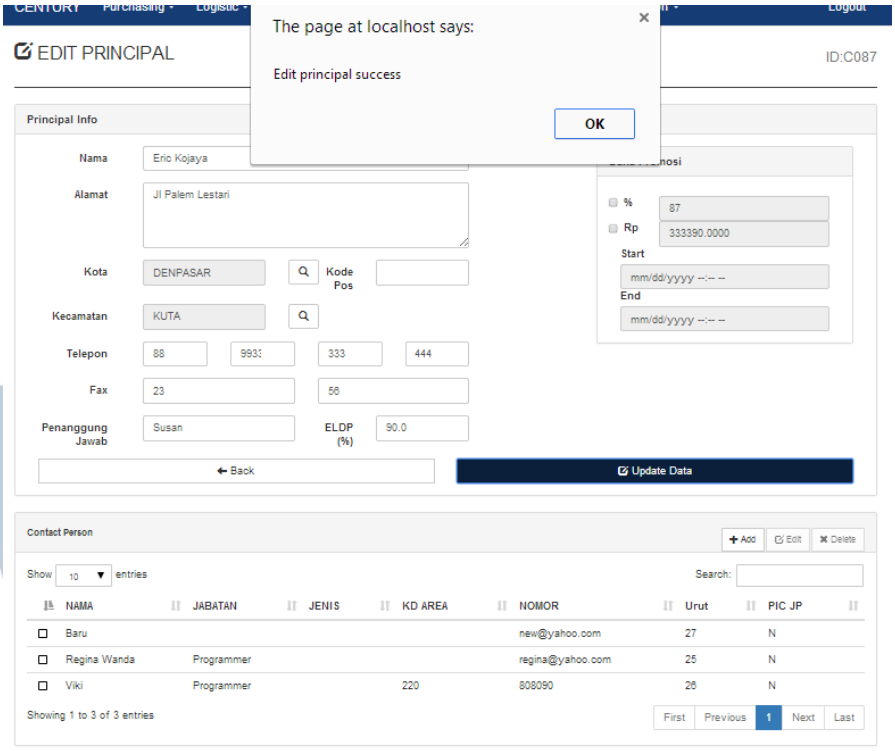
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.65 Halaman Edit_Principal_View

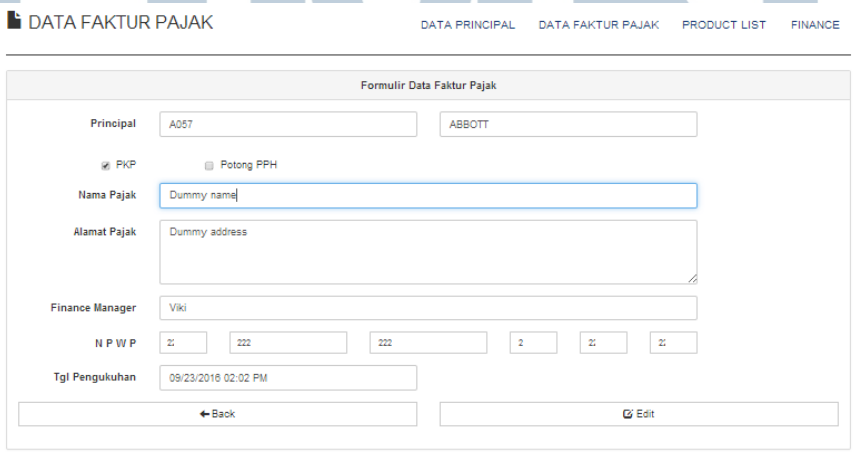
Gambar 3.66 menunjukan dialog notifikasi yang akan dimunculkan apabila user telah selesai melakukan pengubahan pada data principal.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

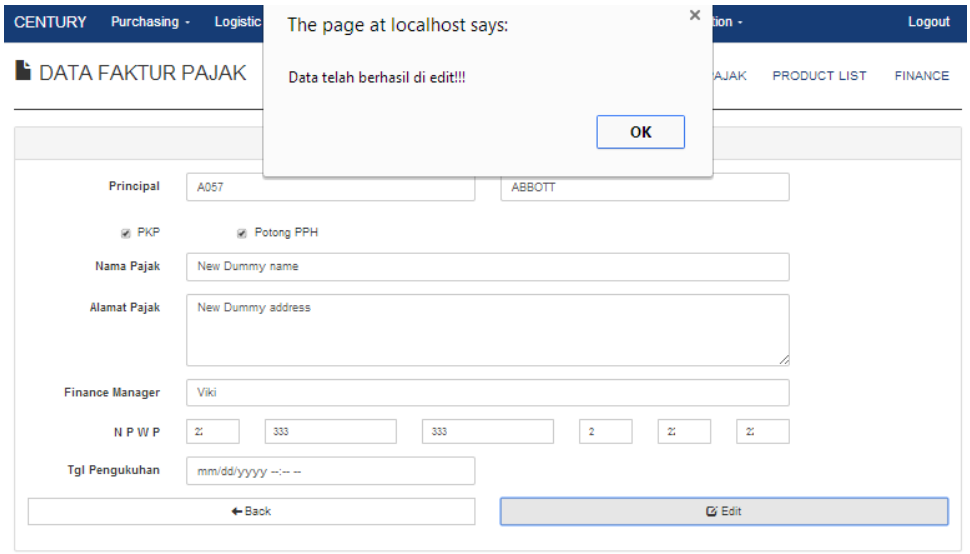


Gambar 3.66 Notifikasi Sukses pada Edit_Principal_View

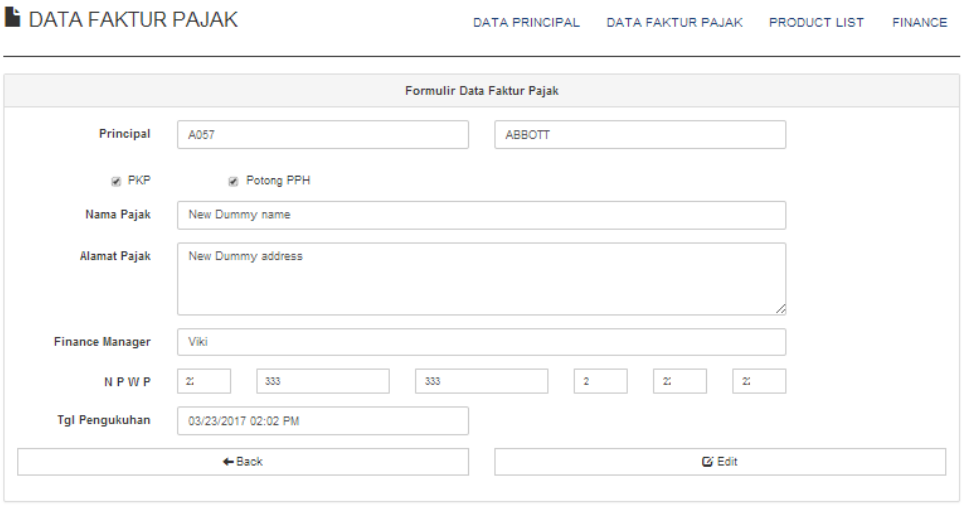
Halaman Faktur_Pajak_View akan menampilkan data faktur milik principal, dimana pada halaman tersebut kita dapat mengubah data faktur yang sudah ada. Setelah melakukan pengubahan maka akan muncul dialog yang memberitahukan pengubahan berhasil dilakukan.



Gambar 3.67 Mengubah Data Faktur Pajak



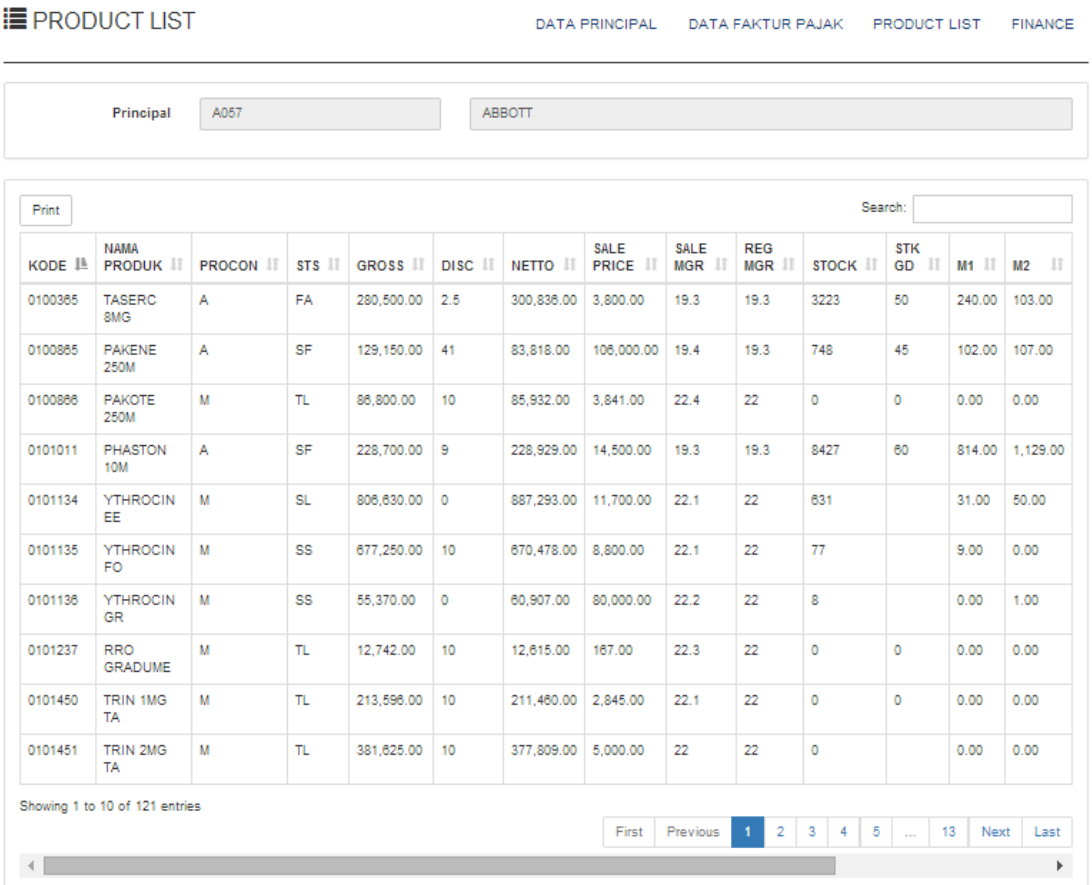
Gambar 3.68 Notifikasi Pengubahan Data Faktur Berhasil



Gambar 3.69 Mengakses Ulang Halaman Faktur Pajak

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Halaman Product_List_View akan menampilkan daftar produk beserta detailnya berdasarkan principal yang dipilih.Berikut adalah gambar untuk halaman daftar produk.

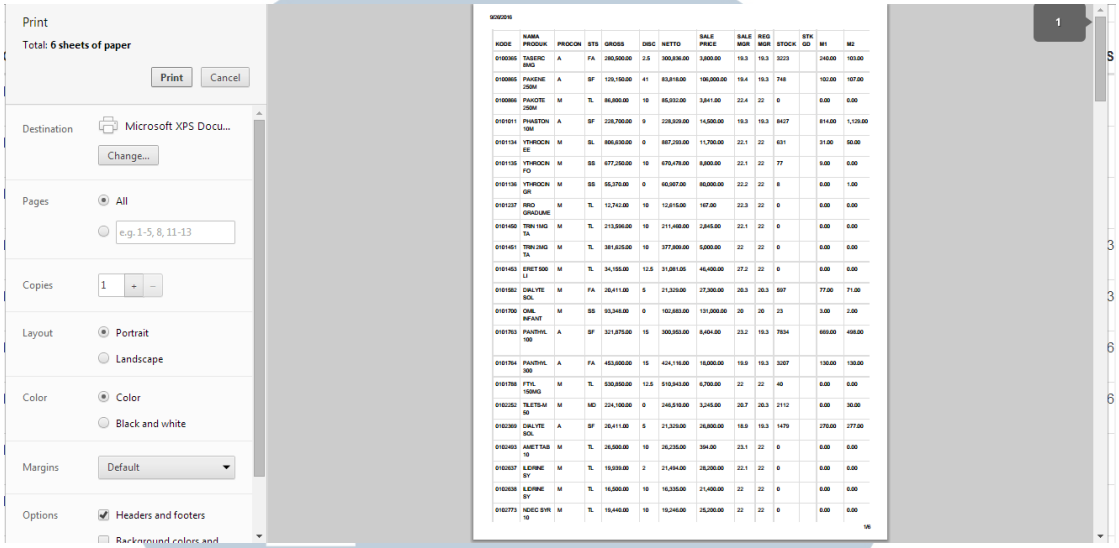


KODE	NAMA PRODUK	PROCON	STS	GROSS	DISC	NETTO	SALE PRICE	SALE MGR	REG MGR	STOCK	STK GD	M1	M2
0100365	TASERC 8MG	A	FA	280,500.00	2.5	300,836.00	3,800.00	19.3	19.3	3223	50	240.00	103.00
0100865	PAKENE 250M	A	SF	129,150.00	41	83,818.00	106,000.00	19.4	19.3	748	45	102.00	107.00
0100866	PAKOTE 250M	M	TL	86,800.00	10	85,932.00	3,841.00	22.4	22	0	0	0.00	0.00
0101011	PHASTON 10M	A	SF	228,700.00	9	228,929.00	14,500.00	19.3	19.3	8427	60	814.00	1,129.00
0101134	YTHROCIN EE	M	SL	806,630.00	0	887,293.00	11,700.00	22.1	22	631		31.00	50.00
0101135	YTHROCIN FO	M	SS	677,250.00	10	670,478.00	8,800.00	22.1	22	77		9.00	0.00
0101136	YTHROCIN GR	M	SS	55,370.00	0	60,907.00	80,000.00	22.2	22	8		0.00	1.00
0101237	RRO GRADUME	M	TL	12,742.00	10	12,615.00	167.00	22.3	22	0	0	0.00	0.00
0101450	TRIN 1MG TA	M	TL	213,596.00	10	211,460.00	2,845.00	22.1	22	0	0	0.00	0.00
0101451	TRIN 2MG TA	M	TL	381,625.00	10	377,809.00	5,000.00	22	22	0		0.00	0.00

Gambar 3.70 Halaman Daftar Produk Principal

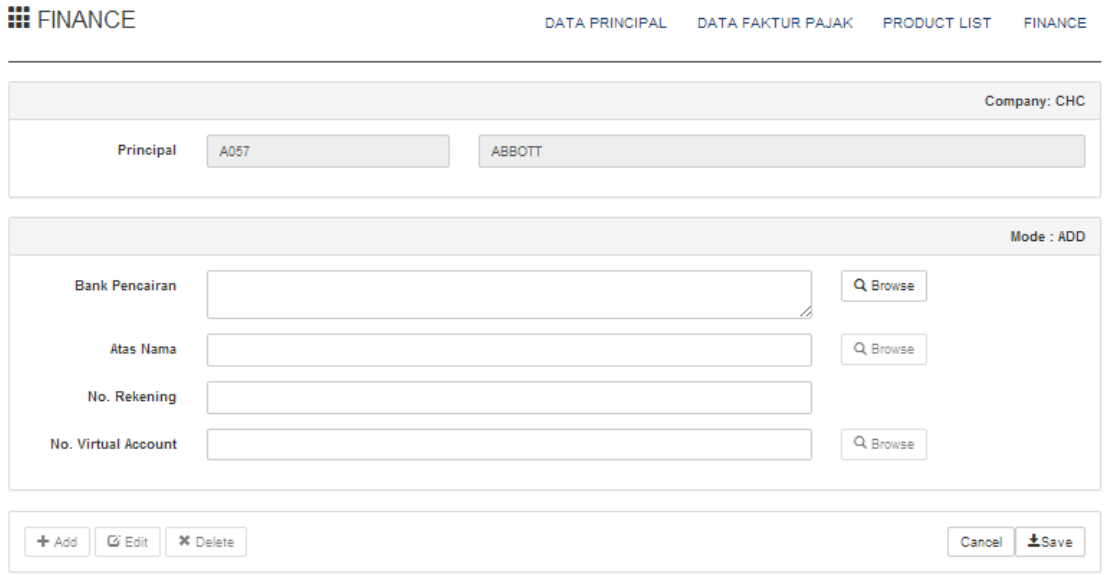
UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Jika user ingin menekan tombol print yang ada pada tabel produk maka halaman print preview akan ditampilkan seperti gambar berikut



Gambar 3.71 Halaman Print Preview Daftar Produk

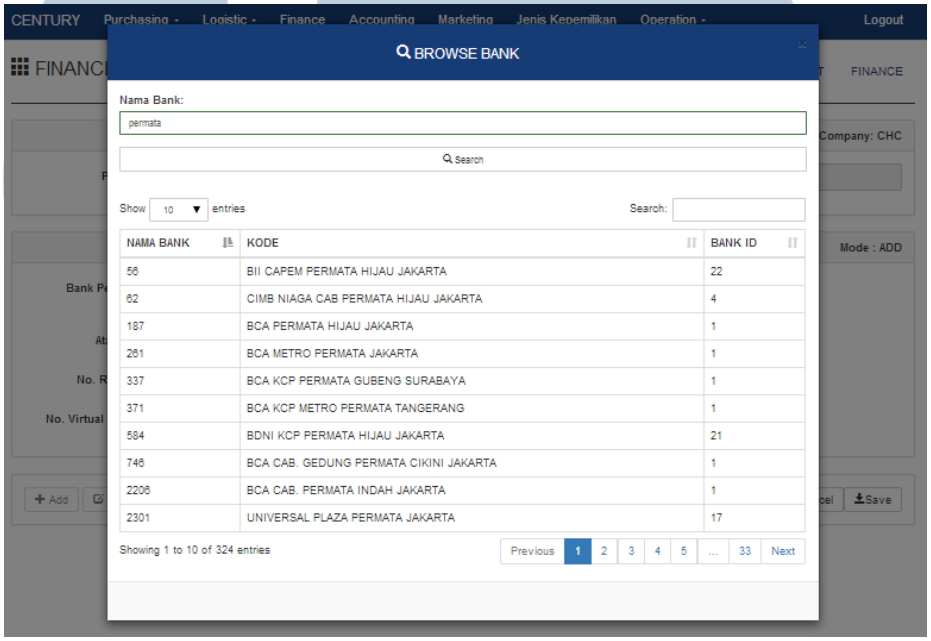
Halaman Finance_View berisi data finance principal ketika diakses pertama kali. Dalam contoh kasus ini principal belum mempunyai data finance sehingga formulir akan kosong. Ketika user menekan tombol Add maka formulir yang tadinya tidak bisa diakses (disabled) menjadi bisa diakses dan Mode akan menjadi ADD.



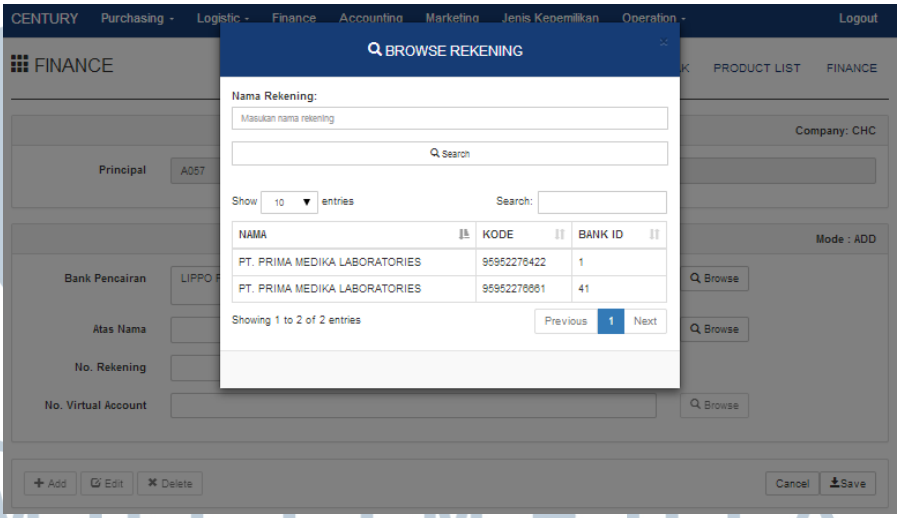
Gambar 3.72 Halaman Finance_View

NUSANTARA

Untuk memasukan nama bank, rekening, virtual account user harus menggunakan tombol browse yang telah disediakan sehingga masing-masing kotak dialog berisi daftar nama bank, rekening, virtual account akan ditampilkan seperti gambar berikut

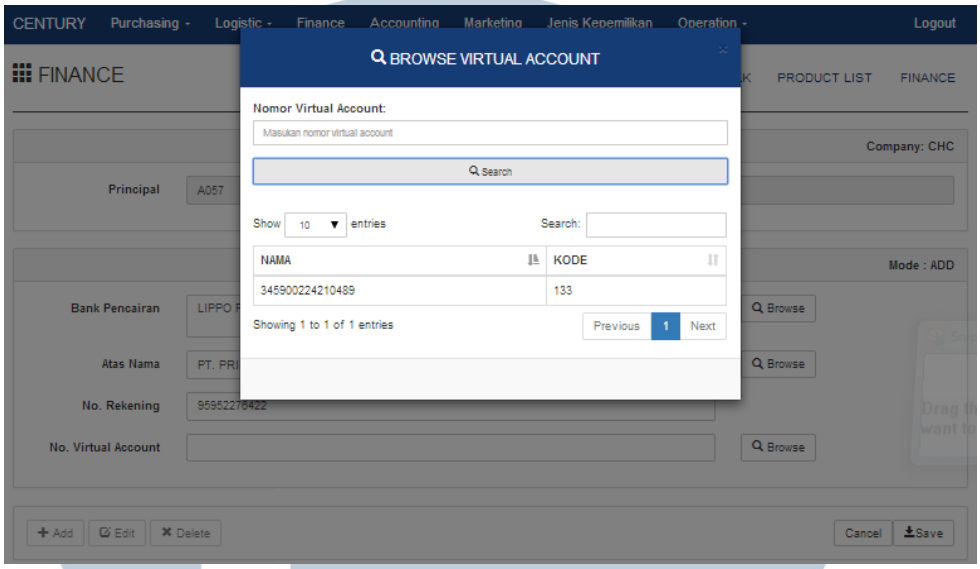


Gambar 3.73 Kotak Dialog Browse Bank

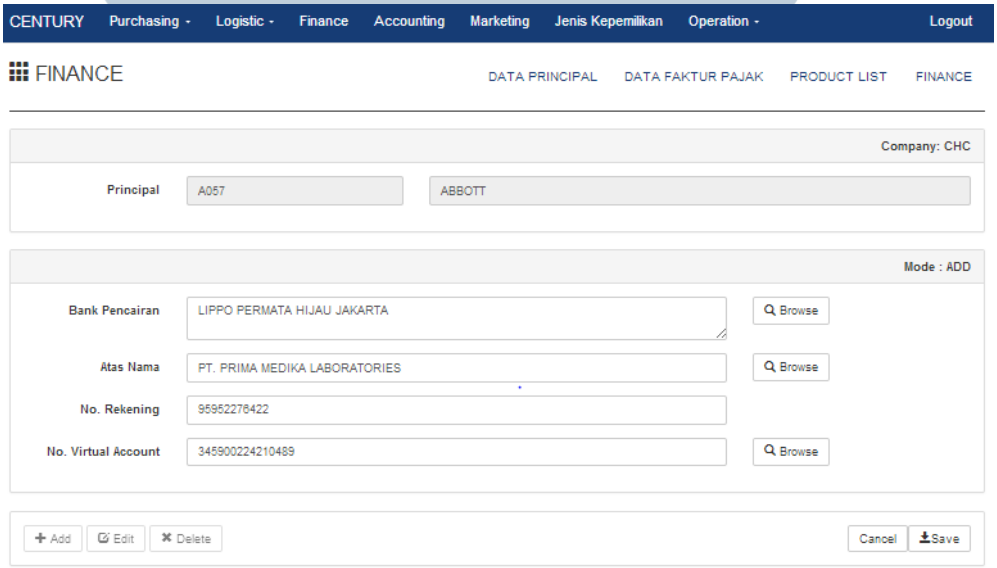


Gambar 3.74 Kotak Dialog Browse Rekening

UNIVERSITAS MULTIMEDIA NUSANTARA

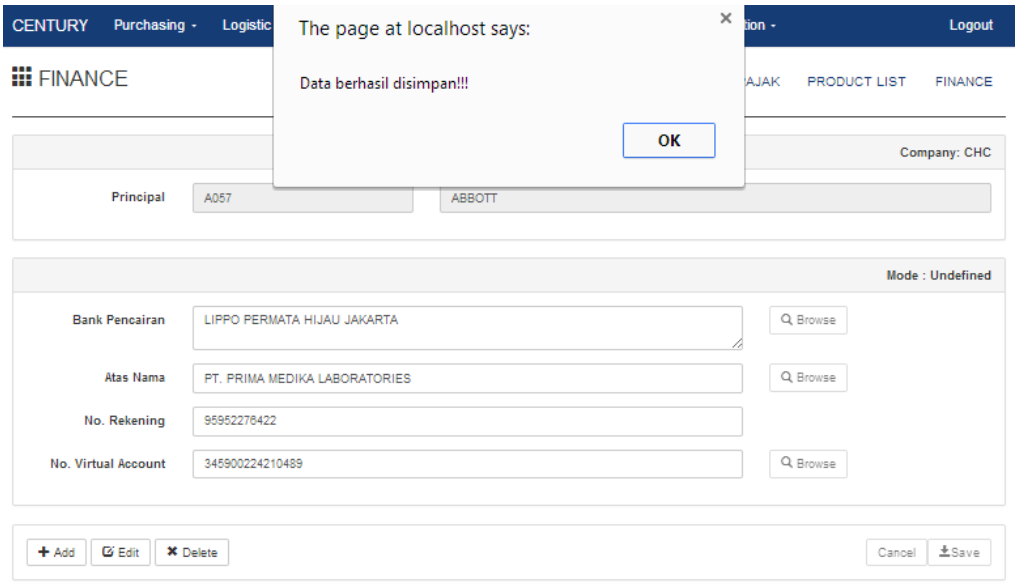


Gambar 3.75 Kotak Dialog Browse Virtual Account



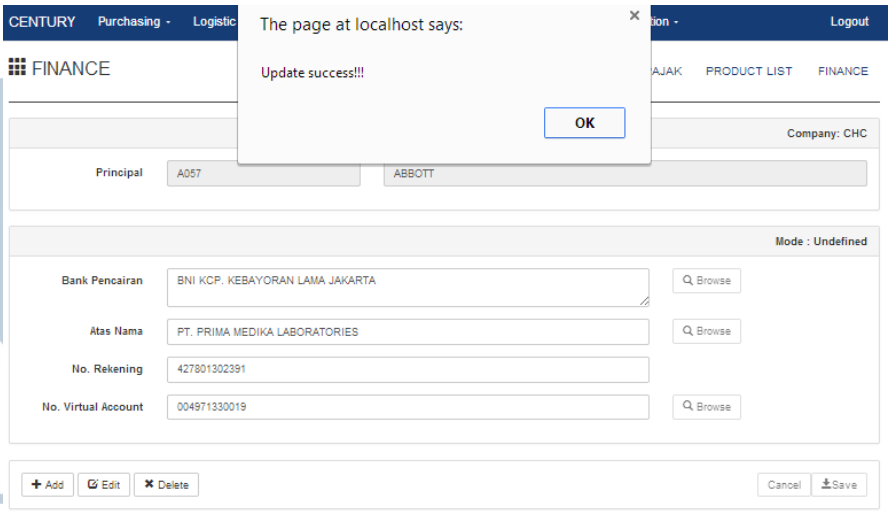
Gambar 3.76 Tampilan Setelah User Memberikan Input Finance

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.77 Notifikasi Data Finance Berhasil Ditambah

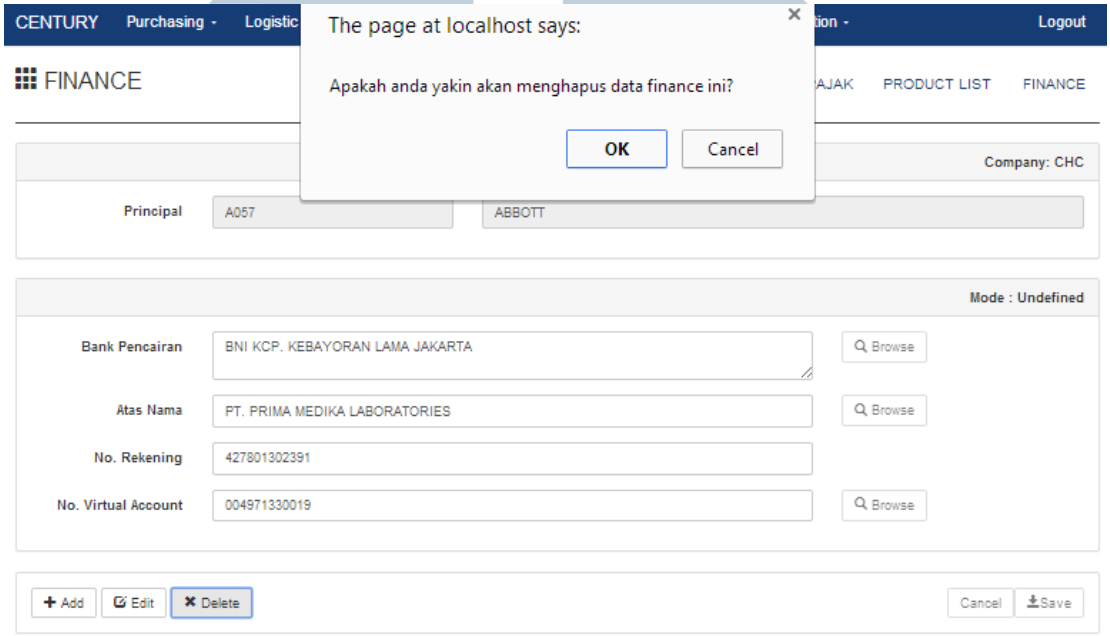
Jika user menekan tombol Edit maka Mode akan berubah statusnya menjadi Edit dan formulir bisa diakses. Jika Edit berhasil dilakukan maka akan muncul kotak dialog notifikasi yang memberitahu bahwa pengubahan data finance berhasil dilakukan.



Gambar 3.78 Notifikasi Data Finance Berhasil Diubah

MULTIMEDIA NUSANTARA

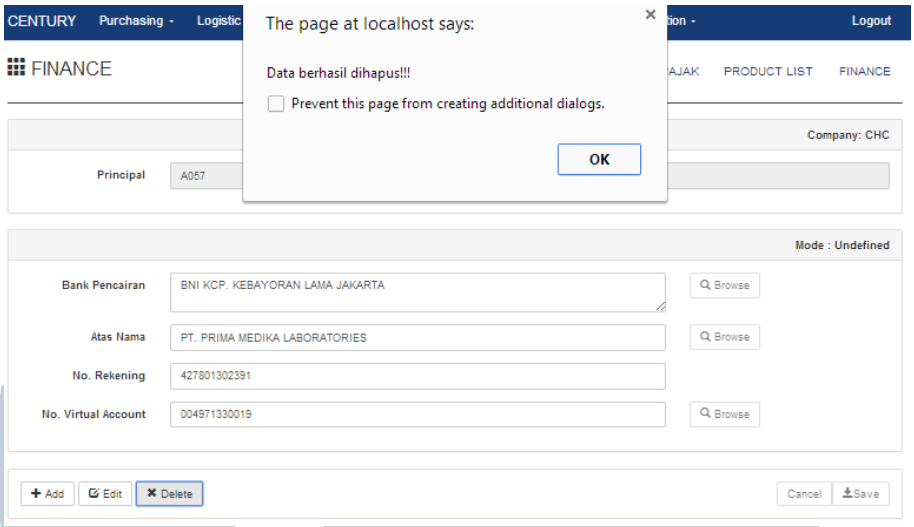
Jika user menekan tombol Delete maka akan muncul kotak dialog konfirmasi untuk meminta konfirmasi hapus dari user seperti tampak pada gambar berikut



Gambar 3.79 Kotak Dialog Konfirmasi Hapus

Jika user memberikan tanggapan positif atas konfirmasi tersebut maka data finance akan menjadi tidak aktif dan kotak dialog notifikasi bahwa data telah berhasil dinoaktifkan akan ditampilkan.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.80 Kotak Dialog Notifikasi Hapus Data Finance

3.3.3 Kendala yang Ditemukan

Kendala yang ditemukan pada pelaksanaan kerja magang adalah sulitnya mendeteksi kesalahan pada modul AngularJS untuk melakukan request ke *controller* CodeIgniter dan tidak adanya akses internet menyebabkan sulitnya untuk menggunakan library-library yang sudah tersedia dan mencari solusi untuk mengatasi kesalahan yang muncul.

3.3.4 Solusi

Dilakukan pencatatan secara mendetail terhadap *software-software* atau *library* yang dibutuhkan dan mengumpulkan *software/library* tersebut pada saat selesai melakukan kegiatan magang untuk dibawa keesokan harinya ke PT Pharos Indonesia untuk meminta izin agar software tersebut boleh dipasang pada komputer milik PT Pharos Indonesia.

Untuk mendeteksi kesalahan pada modul AngularJS, digunakan metode *try* and *catch* yang disediakan AngularJS untuk mendeteksi error yang terjadi.

UNIVERSITAS MULTIMEDIA NUSANTARA