

**SISTEM INFORMASI RESERVASI BARBER BERBASIS WEB  
STUDI KASUS: MARK THE BARBER YOGYAKARTA**

Skripsi



oleh  
**HARTADI KARTANDA**  
72120027

**PROGRAM STUDI SISTEM INFORMASI FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
TAHUN 2016**

**SISTEM INFORMASI RESERVASI BARBER BERBASIS WEB  
STUDI KASUS: MARK THE BARBER YOGYAKARTA**

Skripsi



Diajukan kepada Program Studi Sistem Informasi Fakultas Teknologi Informasi  
Universitas Kristen Duta Wacana  
Sebagai Salah Satu Syarat dalam Memperoleh Gelar  
Sarjana Komputer

Disusun oleh

**HARTADI KARTANDA**  
72120027

**PROGRAM STUDI SISTEM INFORMASI FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS KRISTEN DUTA WACANA  
TAHUN 2016**

---

## HALAMAN PERSETUJUAN

Judul Skripsi : Sistem Informasi Reservasi Barber Berbasis Web  
Studi Kasus: Mark the Barber Yogyakarta  
Nama Mahasiswa : HARTADI KARTANDA  
N I M : 72120027  
Matakuliah : Skripsi  
Kode : SI4046  
Semester : Genap  
Tahun Akademik : 2015/2016

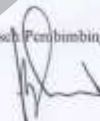
Telah diperiksa dan disetujui di Yogyakarta,  
Pada tanggal 17 Juni 2016

Dosen Pembimbing I



HALIM BUDI SANTOSO, S.Kom., MEd., M.T

Dosen Pembimbing II



BEBI SUTEDIA D. G., S.Kom., M.M.

HALAMAN PENGESAHAN

SISTEM INFORMASI RESERVASI BARBER BERBASIS WEB  
STUDI KASUS: MARK THE BARBER YOGYAKARTA

Oleh: HARTADI KARTANDA / 72120027

Dipertahankan di depan Dewan Penguji Skripsi  
Program Studi Sistem Informasi Fakultas Teknologi Informasi  
Universitas Kristen Duta Wacana - Yogyakarta  
Dan dinyatakan diterima untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana Komputer  
pada tanggal  
16 Juni 2016

Yogyakarta, 17 Juni 2016  
Mengesahkan,

Dewan Penguji:

1. BUDI SUTEDJO D. O., S.Kom., M.M.
2. HALIM BUDI SANTOSO, S.Kom., MBA., MT
3. Drs. JONG JEK SIANG, M.Sc.
4. ERICK KURNILAWAN, S.Kom., M.Kom.



Dekan

(Drs. SUSANTO, S.Kom., M.T.)

Ketua Program Studi

(Drs. JONG JEK SIANG, M.Sc.)

### PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

**Sistem Informasi Reservasi Barber Berbasis Web  
Studi Kasus: Mark the Barber Yogyakarta**

yang saya kerjakan untuk melengkapi sebagian persyaratan menjadi Sarjana Komputer pada pendidikan Sarjana Program Studi Sistem Informasi Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, bukan merupakan tiruan atau duplikasi dari skripsi kesarjanaan di lingkungan Universitas Kristen Duta Wacana maupun di Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Jika dikemudian hari didapati bahwa hasil skripsi ini adalah hasil plagiasi atau tiruan dari skripsi lain, saya bersedia dikenai sanksi yakni pencabutan gelar kesarjanaan saya.

Yogyakarta, 17 Juni 2016



## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah Tuhan Yang Maha Esa, karen berkat limpahan rahmat dan karuniaNya akhirnya penulis dapat menyelesaikan tugas akhir guna memenuhi persyaratan dalam menempuh ujian akhir pada Fakultas Teknologi Informasi Program Sstudi Sistem Informasi Universitas Kristen Duta Wacana. Dalam laporan karya ilmiah ini, dibahas mengenai Sistem Informasi Reservasi Barber Berbasis *Web Responsive* (Studi Kasus Mark The Barber Yogyakarta). Hasil dari penelitian dianalisis berdasarkan teori yang ada untuk kemudian dibuat kesimpulan. Dalam menyelesaikan tugas akhir ini banyak sekali bimbingan, saran dan bantuan yang penulis terima dari berbagai pihak. Untuk itu penulis mengucapkan terima kasih dan penghargaan yang sebesar-besarnya terutama kepada:

1. Tuhan Yang Maha Esa.
2. Kedua orang tua penulis, Bapak Iman Kartanda dan Ibu Aslia yang selalu memberi dukungan dan doa kepada penulis dalam menyelesaikan tugas akhir ini.
3. Halim Budi Santoso, S.Kom., MBA, M.T. selaku Dosen Pembimbing I yang selalu memberikan masukan dan arahan kepada penulis dalam menyelesaikan tugas akhir ini.
4. Budi Sutedjo Dharma Oetomo, S.Kom., M.M. selaku Dosen Pembimbing II yang selalu memberi masukan dan arahan kepada penulis dalam menyelesaikan tugas akhir ini.
5. Mark The Barber Yogyakarta yang bersedia memberikan penulis kesempatan sebagai tempat studi kasus tugas akhir.
6. Aristian Wibowo, Darma Cahyadi, Volney, dan teman-teman penulis yang selalu menemani, membantu, mengingatkan, dan memberikan dukungan kepada penulis dalam menyelesaikan tugas akhir ini.
7. Semua pihak yang membantu penulis atas selesainya laporan ini.

Akhir kata, penulis meminta maaf kepada seluruh pihak apabila ada sikap yang berkenan selama penyelesaian tugas akhir ini. Semoga tugas ini dapat memberikan manfaat sebesar-besarnya kepada semua pihak.

Yogyakarta, 10 Juni 2016

Hartadi Kartanda  
Penulis

©UKDWN

## ABSTRAK

Mark The Barber Yogyakarta yang merupakan perusahaan yang bergerak di bidang cukur rambut, saat ini membutuhkan media promosi yang dapat diakses secara *online* oleh masyarakat luas khususnya Yogyakarta agar mendapatkan informasi tentang layanan yang dimiliki oleh Mark The Barber Yogyakarta.

Pembangunan sistem informasi reservasi pada suatu *barbershop* saat ini sangat diperlukan guna efisien dan efektifitas *barbershop* dalam melayani *customer*. Sistem informasi reservasi *barber* ini mencakup segala informasi mengenai daftar antrian *barberman*, pemesanan tempat, dan informasi-informasi lain. Dengan terus berkembangnya teknologi internet, *web* menjadi suatu alat bantu yang tidak hanya mampu menyediakan informasi, tetapi juga mampu untuk mengolah informasi. Dengan menggunakan *web*, daya jangkau layanan suatu *barbershop* akan sangat luas dan juga memudahkan *customer* tidak perlu datang langsung ke *barbershop* untuk pemesanan tempat.

Proses pengolahan informasi pada sistem informasi reservasi *barber* dengan berbasis *web* menjadi media informasi yang memudahkan *customer* dalam pemesanan tempat. Hadirnya sistem informasi reservasi *barber* pada Mark The Barber Yogyakarta diharapkan dapat memberikan sarana dan kemudahan kepada masyarakat untuk melakukan reservasi tempat secara *online*.

(**Kata kunci:** sistem informasi reservasi *barber*, reservasi *barber*, *barber*, *barbershop*)



## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERNYATAAN KEASLIAN SKRIPSI .....	iv
KATA PENGANTAR .....	vi
ABSTRAK .....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah.....	2
1.4. Spesifikasi Sistem.....	2
1.5. Tujuan dan Manfaat Penelitian.....	3
1.6. Metodologi Penelitian .....	4
1.7. Sistematika Penulisan.....	5
BAB 2 LANDASAN TEORI.....	6
2.1. Sistem .....	6
2.2. Informasi .....	6
2.3. Sistem Informasi.....	6
2.4. Barbershop.....	7
2.5. Reservasi .....	9

2.6.	Struktur Dasar Teori Antrian.....	9
2.7.	Sistem Informasi Reservasi .....	10
BAB 3 ANALISIS DAN PERANCANGAN SISTEM .....		13
3.1.	Analisis Data .....	13
3.2.	Rancangan Proses.....	13
3.2.1.	Data Flow Diagram (DFD) .....	13
3.2.2.	Rancangan Alur Proses .....	16
3.2.3.	Diagram Sequence .....	21
3.2.4.	Diagram Use Case.....	23
3.3.	Model Data Logika (MDL) .....	24
3.4.	Rancangan Antarmuka .....	41
3.4.1	Rancangan Antarmuka <i>Home</i> .....	41
3.4.2	Rancangan Antarmuka Registrasi dan Autentikasi.....	42
3.4.3	Rancangan Antarmuka Reservasi .....	43
3.4.4	Rancangan Antarmuka Pengeluaran Harian .....	45
3.4.5	Rancangan Antarmuka Penggajian <i>Barberman</i> .....	47
BAB 4 IMPLEMENTASI DAN ANALISIS SISTEM.....		49
4.1.	Implementasi Sistem .....	49
4.1.1.	Proses Autentikasi ( <i>Login</i> ).....	49
4.1.2.	Proses Reservasi.....	51
4.1.3.	Pelayanan Reservasi <i>Customer</i> .....	54
4.1.4.	Menampilkan Informasi Gaji <i>Barberman</i> .....	59
4.1.5.	Menampilkan Grafik Tren Rambut dan <i>Barberman</i> .....	60
4.2.	Analisis Sistem .....	61
4.3.	Kelebihan dan Kekurangan Sistem .....	63
4.4.1.	Kelebihan Sistem .....	63

4.4.2. Kekurangan Sistem .....	64
BAB 5 PENUTUP .....	65
5.1. Kesimpulan.....	65
5.2. Saran.....	65
DAFTAR PUSTAKA .....	66
LAMPIRAN A : LISTING PROGRAM .....	66
LAMPIRAN B: DOKUMEN-DOKUMEN TERKAIT.....	67

©UKDWN

## DAFTAR GAMBAR

Gambar 2.1 Struktur Organisasi Mark The Barber .....	7
Gambar 2.2 Proses Bisnis Tanpa Reservasi .....	8
Gambar 2.3 Proses Bisnis Reservasi .....	9
Gambar 2.4 Struktur Dasar Teori Antrian (B & Angwarmasse, 2012) .....	10
Gambar 2.5 Proses Bisnis Reservasi Online .....	11
Gambar 2.6 Proses Reservasi Ditunggalkan .....	12
Gambar 3.1 DFD Level 0 .....	13
Gambar 3.2 DFD Level 1 .....	14
Gambar 3.3 DFD Level 2 Pendaftaran Customer .....	14
Gambar 3.4 DFD Level 2 Proses Reservasi dan Penggajian .....	15
Gambar 3.5 DFD Level 2 Proses Input Pengeluaran .....	15
Gambar 3.6 DFD Level 2 Proses Pencetakan Laporan Administrasi .....	16
Gambar 3.7 Activity Diagram Pendaftaran Customer .....	17
Gambar 3.8 Activity Diagram Proses Reservasi .....	17
Gambar 3.9 Flowchart Proses Reservasi .....	18
Gambar 3.10 Activity Diagram Proses Pengeluaran Harian .....	19
Gambar 3.11 Flowchart Pengeluaran Harian .....	20
Gambar 3.12 Activity Diagram Proses Penggajian Barberman .....	20
Gambar 3.13 Flowchart Penggajian Barberman .....	21
Gambar 3.14 Diagram Sequence Proses Reservasi .....	22
Gambar 3.15 Diagram Sequence Proses Pengeluaran Harian .....	22
Gambar 3.16 Diagram Sequence Proses Penggajian .....	23
Gambar 3.17 Diagram Use Case Mark The Barber .....	24
Gambar 3.18 MDL 1: Menentukan Entitas Utama .....	25

Gambar 3.19 MDL 2: Hubungan Antar Entitas .....	26
Gambar 3.20 MDL 3: Menentukan Kunci Primer (PK) dan Kunci Alternatif (AK) .....	27
Gambar 3.22 MDL 4: Menentukan Kunci Tamu Lanjutan.....	29
Gambar 3.23 MDL 6: Penambahan Atribut Bukan Kunci Lanjutan .....	32
Gambar 3.24 Rancangan Antarmuka Halaman <i>Home</i> .....	41
Gambar 3.25 Rancangan Antarmuka Halaman Autentikasi <i>Customer</i> .....	42
Gambar 3.26 Rancangan Antarmuka Halaman Autentikasi Mark.....	42
Gambar 3.27 Rancangan Antarmuka Halaman Registrasi <i>Customer</i> .....	43
Gambar 3.28 Rancangan Antarmuka Halaman Reservasi .....	44
Gambar 3.29 Rancangan Antarmuka Halaman Detail Reservasi .....	44
Gambar 3.30 Rancangan Antarmuka Halaman Pembelian Produk .....	45
Gambar 3.31 Rancangan Antarmuka Halaman Pengeluaran Harian .....	46
Gambar 3.32 Rancangan Antarmuka Detail Pengeluaran Harian.....	46
Gambar 3.33 Rancangan Antarmuka Halaman Tambah Pengeluaran.....	46
Gambar 3.34 Rancangan Antarmuka Halaman <i>Form Gaji Barberman</i> .....	47
Gambar 3.35 Rancangan Antarmuka Halaman Informasi Gaji <i>Barberman</i> .....	48
Gambar 4.1 Tampilan Halaman Autentikasi.....	50
Gambar 4.2 Tampilan Halaman Utama <i>Customer</i> .....	50
Gambar 4.3 Tampilan Halaman <i>Form</i> Reservasi.....	51
Gambar 4.4 Tampilan Halaman <i>Form</i> Detail Reservasi .....	52
Gambar 4.5 Tampilan Halaman Reservasi Berhasil .....	53
Gambar 4.6 Tampilan Halaman <i>Form</i> Detail Reservasi dengan Pesan Peringatan .....	54
Gambar 4.7 Tampilan Halaman Pelayanan Reservasi .....	54
Gambar 4.8 Tampilan Halaman Pelayanan Reservasi .....	56

Gambar 4.9 Tampilan Halaman Detail Reservasi.....	56
Gambar 4.10 Tampilan Halaman <i>Form</i> Belanja Produk .....	57
Gambar 4.11 Tampilan Halaman Detail Reservasi.....	58
Gambar 4.12 Tampilan Halaman Pelayanan Reservasi .....	59
Gambar 4.13 Tampilan Halaman <i>Form</i> Informasi Gaji <i>Barberman</i> .....	59
Gambar 4.14 Tampilan Halaman Informasi Gaji <i>Barberman</i> .....	60
Gambar 4.15 Tampilan Grafik Tren Gaya Rambut dan <i>Barberman</i> .....	61

©UKDW

## DAFTAR TABEL

Tabel 2.1 Aktivitas <i>Stakeholder</i> Mark The Barber .....	8
Tabel 3.1 Aturan bisnis Reservasi.....	30
Tabel 3.2 Aturan Bisnis Pengeluaran & Penggajian .....	30
Tabel 3.3 Kamus Data <i>Customer</i> .....	37
Tabel 3.4 Kamus Data Karyawan .....	37
Tabel 3.5 Kamus Data <i>Treatment</i> .....	38
Tabel 3.6 Kamus Data Gaya Rambut.....	38
Tabel 3.7 Kamus Data Produk .....	38
Tabel 3.8 Kamus Data Libur.....	38
Tabel 3.9 Kamus Data Lembur .....	39
Tabel 3.10 Kamus Data Gaji Barberman .....	39
Tabel 3.11 Kamus Data Pengeluaran .....	39
Tabel 3.12 Kamus Data Detail Pengeluaran .....	39
Tabel 3.13 Kamus Data Layanan Pengeluaran .....	39
Tabel 3.14 Kamus Data Penggajian .....	40
Tabel 3.15 Kamus Data Reservasi .....	40
Tabel 3.16 Kamus Data <i>R_Treatment</i> .....	40
Tabel 3.17 Kamus Data <i>R_Produk</i> .....	40
Tabel 4.1 Data Transaksi Mark The Barber Manual.....	62
Tabel 4.2 Tabel Data Transaksi Mark The Barber Sistem.....	63

## DAFTAR LAMPIRAN

Controller <i>Home</i> .....	A 1
Controller Autentikasi .....	A 1
Controller Register .....	A 2
Controller Akun .....	A 3
Controller Reservasi .....	A 5
Controller Produk .....	A 8
Controller Treatment .....	A 10
Controller Gaya Rambut .....	A 12
Controller Lembur .....	A 14
Controller Libur .....	A 15
Controller Gaji .....	A 16
Controller Penggajian .....	A 17
Controller Laporan .....	A 18
Model Autentikasi .....	A 20
Model Customer .....	A 20
Model Gaji .....	A 21
Model Gaya Rambut .....	A 22
Model Karyawan .....	A 22
Model Laporan .....	A 23
Model Lembur .....	A 24
Model Libur .....	A 25
Model Pengeluaran .....	A 25
Model Penggajian .....	A 26
Model Produk .....	A 27



Model Register .....	A 28
Model Reservasi.....	A 28
Model Treatment.....	A 30
View Head CSS .....	A 31
View Head Menu .....	A 31
View Footer.....	A 32
View Javascript.....	A 33
Gambar Data Transaksi Mark The Barber (1) .....	B 1
Gambar Data Transaksi Mark The Barber (2) .....	B 1
Gambar Data Transaksi Sistem Mark The Barber.....	B 2

©UKYDWN

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang Masalah

Perkembangan teknologi yang sangat pesat menuntut kita untuk mengenal dan memanfaatkan teknologi. Saat ini, baik individu maupun perusahaan serta lembaga-lembaga membutuhkan teknologi dalam melakukan aktivitas sehari-hari. Teknologi sendiri dapat meringankan beban pekerjaan manusia karena sistem terkomputerisasi dapat memudahkan dan mempercepat pekerjaan manusia. Selain itu, sistem terkomputerisasi dapat mengurangi kesalahan-kesalahan yang mungkin terjadi.

Lokasi yang menjadi studi kasus adalah Mark The Barber Yogyakarta yang beralamat di Jalan Kaliurang, gang Pandega Karya, no. 290, Depok, Kecamatan Sleman, Daerah Istimewa Yogyakarta. Mark The Barber merupakan sebuah tempat penyedia jasa potong atau cukur rambut pria yang berada di Yogyakarta. Saat ini, Mark The Barber memiliki lima layanan, yaitu *Premium Barber Cut*, *Dry Barber Cut*, *Kids Premium Barber Cut*, *Traditional Shaving*, dan *Hair Styling*. Mark The Barber memiliki empat orang *barberman* yang siap melayani *customer* baik yang melakukan reservasi maupun yang datang secara langsung.

Proses reservasi di Mark The Barber saat ini masih dilakukan secara manual. Kasir Mark masih memanfaatkan media sosial, *sms* maupun aplikasi pesan singkat dalam proses reservasi. *Customer* yang ingin melakukan reservasi wajib memberikan informasi data diri ke kasir melalui komunikasi media sosial atau *sms*.

Masalah yang dihadapi oleh Mark The Barber adalah waktu reservasi serta jumlah *customer* yang melakukan reservasi. Pihak *barbershop* sulit memprioritaskan *customer* yang melakukan reservasi dan *customer* yang datang tanpa melakukan reservasi terlebih dahulu. Hal ini fatal, karena akan menimbulkan ketidaknyamanan pada *customer* baik yang datang secara langsung maupun yang melakukan reservasi.

Untuk menyelesaikan masalah di atas, maka akan dibangun sebuah sistem informasi yang dapat membantu menangani masalah reservasi pada Mark The Barber. Sistem ini akan mempermudah dalam mencatat data reservasi *customer* serta memiliki batasan waktu reservasi sehingga dapat membantu pihak *barbershop* dalam memprioritaskan *customer*.

### 1.2. Rumusan Masalah

Permasalahan yang menjadi fokus dalam penelitian ini adalah:

- a. Bagaimana penerapan metode reservasi pada *barbershop*?

### 1.3. Batasan Masalah

Berdasarkan rumusan masalah di atas, maka batasan-batasan masalah dalam penelitian ini adalah:

- a. Sistem dibangun untuk proses reservasi online.
- b. Sistem hanya ditujukan untuk *customer* yang telah mendaftar sebagai member.
- c. Sistem hanya melayani reservasi dengan batasan kuota *barberman* yang telah ditentukan.
- d. Reservasi dikatakan gagal apabila *customer* tidak berada di *barbershop* dalam waktu selambat-lambatnya 15 menit dari waktu yang telah ditentukan.

### 1.4. Spesifikasi Sistem

Spesifikasi sistem yang dibangun terbagi dalam 5 hal, yaitu:

1. Spesifikasi aplikasi/ program
  - a. Program mampu mengelola data *customer*, karyawan, *treatment* (layanan), produk, reservasi, gaji *barberman*, libur *barberman*, dan lembur *barberman*.
  - b. Program mampu menyajikan informasi antrian reservasi berdasarkan *barberman*.
  - c. Program mampu menyajikan (rekomendasi) jenis-jenis potongan rambut yang sedang tren.
  - d. Program mampu menampilkan informasi gaji *barberman*.

- e. Program mampu menutup sistem reservasi.
  - f. Program mampu membatasi waktu dan kuota reservasi.
  - g. Program mampu mengelola data pengeluaran *barbershop*.
2. Spesifikasi perangkat lunak
    - a. Sistem Operasi Windows 7 Professional Service Pack 1.
    - b. XAMPP Win32-5.6.12-0-VC11 lengkap dengan PHP.
    - c. Sublime Text 3.
    - d. Browser: Maxthon Cloud Browser (Rekomendasi), Chrome, Mozilla Fire Fox.
  3. Spesifikasi perangkat keras
    - a. Intel Core 2 Quad Processor Q8300 (4M Cache, 2.50 GHz, 1333 MHz FSB).
    - b. RAM 2GB.
    - c. HDD 80GB.
    - d. VGA Inter HD Inside.
    - e. Monitor, resolusi 1024x768.
    - f. *Keyboard dan Mouse.*
  4. Spesifikasi kecerdasan pembangun
    - a. Kemampuan dalam penggunaan bahasa pemrograman PHP dan SQL.
    - b. Memahami istilah-istilah dalam bahasa pemrograman berbasis web.
    - c. Kemampuan menggunakan aplikasi pengolahan data dan kata untuk membuat laporan dan mendukung analisis data.
  5. Spesifikasi kecerdasan pengguna aplikasi
    - a. Mampu mengoperasikan komputer dan mengakses internet.
    - b. Mampu menggunakan *web browser*.
    - c. Memahami istilah-istilah umum dalam aplikasi *web*.

### **1.5. Tujuan dan Manfaat Penelitian**

Penelitian pada kasus reservasi Mark The Barber ini bertujuan untuk:

1. Membangun dan menghasilkan sebuah sistem yang dapat membantu proses bisnis dan penggajian di Mark The Barber, Yogyakarta.

2. Meningkatkan pelayanan *barber* dan meningkatkan keuntungan *barber* dengan bertambahnya pelanggan yang melakukan reservasi pada saat hari libur atau waktu tertentu.

### 1.6. Metodologi Penelitian

Dalam melakukan penelitian ini, penulis mempergunakan beberapa metode dalam proses pengumpulan data, antara lain:

- a. Wawancara

Suatu bentuk metode riset dengan mengajukan beberapa pertanyaan terhadap orang yang mempunyai peran penting pada objek penelitian penulis. Penulis melakukan wawancara secara langsung kepada pihak-pihak yang berkaitan (pemilik) dengan sistem reservasi Mark The Barber.

- b. Observasi

Proses pengamatan objek penelitian secara langsung di lapangan pada saat melakukan riset. Penulis melakukan pengamatan langsung ke perusahaan tersebut untuk mendapatkan segala sesuatu yang ada kaitannya dengan kegiatan pelayanan untuk mengetahui masalah yang terjadi.

- c. Studi Pustaka

Metode ini dilakukan dengan mengumpulkan data melalui berbagai referensi buku, internet dan catatan kuliah yang digunakan sebagai acuan terhadap kasus yang ditemui.

- d. Analisis Sistem

Analisis sistem yang dilakukan meliputi analisis kebutuhan fungsional dan kebutuhan non fungsional.

- e. Perancangan Sistem

Perancangan sistem meliputi perancangan basis data, perancangan proses (UML), perancangan struktur menu dan perancangan antarmuka.

- f. Implementasi

Mengimplementasikan hasil perancangan sistem menjadi sebuah aplikasi dengan cara *coding* dan *testing*.

### 1.7. Sistematika Penulisan

Penulisan laporan tugas akhir ini akan disusun dalam lima bab dengan sistematika penulisan sebagai berikut:

Bab 1 Pendahuluan berfungsi untuk memberikan gambaran umum tentang penelitian. Bab ini berisi sub-bab latar belakang, rumusan masalah, batasan masalah, spesifikasi sistem, tujuan penelitian, metode penelitian dan sistematika penulisan laporan. Selanjutnya pada Bab 2 terdiri dari tinjauan pustaka dan landasan teori yang membahas konsep-konsep yang diperlukan sebagai dasar penelitian tugas akhir.

Bab 3 merupakan bab perancangan sistem yang akan membahas bagaimana sumber-sumber teori dari bab sebelumnya diimplementasikan pada sistem yang akan dibuat dan rancangan kerja sistem serta alur atau proses-proses yang digunakan dalam pembuatan sistem. Bab 4 merupakan bab penerapan sistem berdasarkan rancangan dan proses-proses yang digunakan dalam pembuatan sistem dari perancangan sistem pada bab 3.

Bab terakhir yaitu bab 5 berisi kesimpulan dari analisis yang dilakukan serta hasil penelitiannya, serta saran-saran untuk kegiatan riset ke depan. Kesimpulan juga berisi jawaban dari pertanyaan penelitian yang dinyatakan dalam rumusan masalah.

## **BAB 5**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil penelitian dan analisa pada sistem yang dibangun, maka dapat disimpulkan beberapa hal sebagai berikut:

- a. Sistem Informasi Reservasi *Barbershop* Berbasis *Web Responsive* dapat membantu Mark The Barber dalam mencatat data reservasi.
- b. Fitur batasan jam dan kuota dapat mencegah *overload* reservasi sehingga dapat memprioritaskan *customer* yang datang secara langsung.
- c. Fitur filter ketersediaan *barberman* membantu mengantisipasi terjadinya kesalahan pemilihan *barberman* pada saat reservasi.

#### **5.2. Saran**

Keterbatasan kemampuan yang dimiliki oleh pembangun. Sistem Informasi *Barbershop* masih memiliki kekurangan dan kelemahan. Adapun saran pengembangan sistem yang dibuat antara lain:

- a. Sistem ini dapat dikembangkan dengan basis *mobile* sehingga memudahkan *customer* dalam mengakses sistem kapan saja.
- b. Sistem dapat ditambahkan fitur notifikasi jadwal reservasi *customer* melalui *e-mail* atau dalam sistem berbasis *mobile*.
- c. Sistem dapat diberi fasilitas belanja produk secara online melalui *website barbershop*.

## DAFTAR PUSTAKA

- Aryani, D., Setiadi, A., Adnandi, M. A., Nurjaman, I., & Widiarso, R. T. (2014, Desember 6). Perancangan Sistem Informasi Pelayanan Klinik Menggunakan Model Antrian First In First Out. *Seminar Nasional Informatika Medis (SNIMed)*, *V*, 22.
- B, J. E., & Angwarmasse, W. (2012). Model Antrian FIFO (First In First Out) Pada Pelayanan Mahasiswa Fakultas Teknik Universitas Janabadra Berbasis Multimedia. *Jurnal Teknik*, *2*.
- Christanto, W., Prasida, A. S., & Fibriani, C. (2012, Januari). Perancangan dan Implementasi Sistem Reservasi Foodcourt Berbasis Web dengan Memanfaatkan Koneksi Wifi. *Jurnal Buana Informatika*, *3*, 39-50.
- Fahrudin, A., Purnama, B. E., & Riasti, B. K. (2011). Pembangunan Sistem Informasi Layanan Haji Berbasis Web. *Journal Speed – Sentra Penelitian Engineering dan Edukasi*, *3*, 36.
- Wibowo. 2014. “Rancang Bangun Sistem Informasi Reservasi Sewa Kamar Hotel Berbasis Web (Studi Kasus Hotel Green Mandarin Pekalongan)”. *Skripsi*. Fakultas Teknologi Informasi, Program Studi Teknik Informatika, Universitas STIKUBANK (UNISBANK) Semarang.
- Yuliawan, Y., Sunarto, M. D., & Soebijono, T. (2013). Pengembangan Sistem Informasi Pendataan Jemaat Gereja Masehi Advent Hari Ketujuh Konferens Jawa Kawasan Timur Berbasis Web. *JSIKA*, *2*, 85. Retrieved from <http://jurnal.stikom.edu/index.php/jsika>



**LAMPIRAN A : LISTING PROGRAM**

## Controller Home

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Home extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->load->model('m_customer','customer',TRUE);
$this->load->model('m_karyawan','karyawan',TRUE);
$this->load->model('m_produk','produk',TRUE);
$this->load->model('m_reservasi','reservasi',TRUE);

public function index(){
$customerID = $this->session->userdata('customerID');
//ambil data customer
$data = array(
'user' => $this->customer->GetCustomerByID($customerID),
'produk' => $this->produk->GetAllProduk(),
'ryan' => $this->reservasi->GetReservasiRyan(),
'heri' => $this->reservasi->GetReservasiHeri(),
'pandu' => $this->reservasi->GetReservasiPandu(),
'dayat' => $this->reservasi->GetReservasiDayat(),
'status' => $this->reservasi->CekStatus(),
'tanggal' => tgl_indo(date('Y-m-d')),
'title' => "Home - Mark The Barber");
$this->load->view('home', $data);}}?>
```

## Controller Autentikasi

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Autentikasi extends CI_Controller{
function __construct(){
parent::__construct();
$this->load->model('m_authentikasi','auth',TRUE);
$this->load->model('m_customer','customer',TRUE);
$this->load->model('m_reservasi','reservasi',TRUE);

public function index(){
if($this->session->userdata('logged_in')){
redirect(base_url());}else{
$this->form_validation->set_error_delimiters('<div class="error">','</div>');
$this->form_validation->set_rules('email','Email','trim|required|xss_clean');
$this->form_validation->set_rules('password','Password','trim|required|xss_clean|callback_authentication');
if($this->form_validation->run() == TRUE){
//Go to private area/ login
$data = array(
'customerID' => $this->input->post('customerID'),
'email' => $this->input->post('email'),
'logged_in' => TRUE);
$this->session->set_userdata($data);
redirect(base_url(), $data);}else{
$data = array(
'status' => $this->reservasi->CekStatus(),
'title' => "Autentikasi - Mark The Barber");
$this->load->view('otentikasi', $data);}}}

public function authentication($password){
//Field validation succeeded. Validate against database
$customerID = $this->input->post('customerID');
$email = $this->input->post('email');
//query the database
$result = $this->auth->login_member($customerID, $email, md5($password));
if($result){
$sess_array = array();
foreach($result as $row){
$sess_array = array(
'customerID' => $row->customerID,
'email' => $row->email);
$this->session->set_userdata('logged_in', $sess_array);}
return TRUE;}else{
$this->form_validation->set_message('authentication','Kesalahan pada Email atau Password');
```

```

return false;}}

public function logout() {
$this->session->sess_destroy();
redirect('autentikasi');}}?>

```

## Controller Register

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Register extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->load->model('m_register','register',TRUE);
$this->load->model('m_customer','customer',TRUE);
$this->load->model('m_reservasi','reservasi',TRUE);

function index(){
$this->form_validation->set_error_delimiters('<div class="input-required">',
'</div>');
$this->form_validation->set_rules('email', 'E-mail', 'required|valid_email');
//cek form validasi
if ($this->form_validation->run() == TRUE){
$email = $this->input->post('email');
$hasil = $this->customer->CekEmail($email);
if($hasil->num_rows() == 0){
//arahkan masuk form pendaftaran akun customer
$data = array(
'email' => $email,
'title' => "Autentikasi - Mark The Barber");
$this->load->view('daftar_akun', $data);}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Email sudah digunakan atau Gunakan email yang
valid</strong></div>");
header('location:'.base_url().'autentikasi');}}else{
$data = array(
'title' => "Autentikasi - Mark The Barber");
$this->load->view('autentikasi', $data);}

function daftarakun(){
//ambil data yang di post
$customerID = $this->input->post('customerID');
$namaDepan = $this->input->post('namaDepan');
$namaBelakang = $this->input->post('namaBelakang');
$jenisKelamin = $this->input->post('jenisKelamin');
$email = $this->input->post('email');
$password = $this->input->post('password');
$data = array(
'customerID' => $customerID,
'namaDepan' => $namaDepan,
'namaBelakang' => $namaBelakang,
'jenisKelamin' => $jenisKelamin,
'alamat' => NULL,
'telp' => NULL,
'email' => $email,
'password' => md5($password),
'tglBuat' => date('Y-m-d H:i:s'),
'tglUbah' => NULL,
'aktif' => 0,
'level' => NULL);
//tambahkan data akun ke database
$id = $this->register->TambahAkun($data);
if($id == 1){
//enkripsi id
//$encrypted_id = md5($id);
$enkripsi = md5($customerID);
//memanggil library email dan set konfigurasi untuk pengiriman email
$this->load->library('email');
//$config = array();
$config['charset'] = 'utf-8';
$config['useragent'] = 'Codeigniter';
$config['protocol'] = "smtp";
$config['mailtype'] = "html";

```

```

$config['smtp_host'] = "ssl://smtp.gmail.com";//pengaturan smtp
$config['smtp_port'] = "465";
$config['smtp_timeout'] = "400";
$config['smtp_user'] = "hartadikartanda.developer@gmail.com"; // isi dengan email
kamu
$config['smtp_pass'] = "skuteromatik"; // isi dengan password kamu
$config['crlf'] = "\r\n";
$config['newline'] = "\r\n";
$config['wordwrap'] = TRUE;
$this->email->initialize($config);
//konfigurasi pengiriman
$this->email->from('hartadikartanda.developer@gmail.com','Hartadi Kartanda');
$this->email->to($email); //email pendaftar yang akan dikirim email verifikasi
$this->email->subject("Verifikasi Akun");
$this->email->message(
"Terima Kasih $namaDepan $namaBelakang telah melakukan registrasi, untuk
mengaktifkan akun anda silahkan klik tautan dibawah ini<br><br>".
site_url("register/verifikasiakun/".$$enkripsi));
if($this->email->send()){
$data = array(
'status' => $this->reservasi->CekStatus(),
'nama' => $namaDepan,
'title' => "Registrasi Sukses - Mark The Barber");
$this->load->view('daftar_akun_sukses', $data); //Tampilkan halaman berhasil
registrasi
//echo "Berhasil melakukan registrasi, silahkan cek email kamu";}
else{
//$this->load->view(); //Tampilkan halaman berhasil registrasi, tetapi gagal
mengirim email verifikasi
echo "Berhasil melakukan registrasi, namu gagal mengirim verifikasi email";}
}else{
//gagal register
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>GAGAL melakukan register akun baru!</strong></div>");
header('location:'.base_url().'register');}}

public function verifikasiakun($key){
$this->register->UbahAktivasi($key);
//Tampilkan halaman berhasil verifikasi akun
$data = array(
'status' => $this->reservasi->CekStatus(),
'title' => "Verifikasi Sukses - Mark The Barber");
$this->load->view('verifikasi_sukses', $data);}}?>

```

### Controller Akun

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Akun extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->cek_login();
$this->load->model('m_customer','customer',TRUE);
$this->load->model('m_reservasi','reservasi',TRUE);
$this->load->model('m_treatment','treatment',TRUE);
$this->load->model('m_produk','produk',TRUE);
$this->load->model('m_gayarambut','gayarambut',TRUE);
$this->load->model('m lembur','lembur',TRUE);
$this->load->model('m_gaji','gaji',TRUE);}
private function cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'backend');}}

public function profile(){
$customerID = $this->session->userdata('customerID');
$data = array(
'user' => $this->customer->GetCustomerByID($customerID),
'status' => $this->reservasi->CekStatus(),
'title' => "Profile - Mark The Barber");
$this->load->view('akun/personal_akun', $data);}

public function history(){
$customerID = $this->session->userdata('customerID');

```

```

$data = array(
    'user' => $this->customer->GetCustomerByID($customerID),
    'reservasi' => $this->customer->GetRiwayatReservasi($customerID),
    'status' => $this->reservasi->CekStatus(),
    'title' => "Riwayat Reservasi - Mark The Barber";
    $this->load->view('akun/history', $data);

public function detail_history($kodeReservasi){
    $data['list_riwayat'] = $this->customer->GetDetailRiwayat($kodeReservasi);
    //echo json_encode($data);
    $this->load->view('akun/detail_history', $data);}

public function update(){
    //form validasi
    $this->form_validation->set_error_delimiters('<div class="input-required">',
    '</div>');
    $this->form_validation->set_rules('namaDepan', 'Nama Depan', 'required');
    $this->form_validation->set_rules('email', 'Email', 'required');
    // $this->form_validation->set_rules('telp', 'Telp', 'required');
    //cek form validasi
    if ($this->form_validation->run() == TRUE){
        $customerID = $this->input->post('customerID');
        $email = $this->input->post('email');
        $namaDepan = $this->input->post('namaDepan');
        $namaBelakang = $this->input->post('namaBelakang');
        $alamat = $this->input->post('alamat');
        $telp = $this->input->post('telp');
        $data = array(
            'email' => $email,
            'namaDepan' => $namaDepan,
            'namaBelakang' => $namaBelakang,
            'alamat' => $alamat,
            'telp' => $telp,
            'tglUbah' => date("Y-m-d H:i:s"));
        $hasil = $this->customer->Update($data, array('customerID' => $customerID));
        if($hasil == 1){
            $this->session->set_flashdata("sukses", "<div class='alert alert-
            success'><strong>Update data BERHASIL dilakukan</strong></div>");
            header('location:'.base_url().'.profile');}else{
            $this->session->set_flashdata("alert", "<div class='alert alert-
            danger'><strong>Update data GAGAL di lakukan</strong></div>");
            header('location:'.base_url().'.profile');}else{
            $customerID = $this->session->userdata('customerID');
            $data = array(
                'user' => $this->customer->GetCustomerByID($customerID),
                'title' => "Profile - Mark The Barber";
            $this->session->set_flashdata("alert", "<div class='alert alert-
            danger'><strong>Update data GAGAL di lakukan</strong></div>");
            header('location:'.base_url().'.profile');}}

public function updatepass(){
    //form validasi
    $this->form_validation->set_error_delimiters('<div class="input-required">',
    '</div>');
    $this->form_validation->set_rules('customerID', 'ID', 'required');
    $this->form_validation->set_rules('passwordLama', 'Password Lama', 'required');
    $this->form_validation->set_rules('password', 'Password Baru',
    'required|min_length[8]|max_length[15]');
    $this->form_validation->set_rules('passconf', 'Konfirmasi',
    'required|matches[password]');
    //cek form validasi
    if ($this->form_validation->run() == TRUE){
        $customerID = $this->input->post('customerID');
        $password = $this->input->post('passwordLama');
        $passwordLama = md5($password);
        $result = $this->customer->GetByPass($customerID, $passwordLama);
        if($result == 1){
            $customerID = $this->input->post('customerID');
            $passwordBaru = $this->input->post('password');
            $data = array(
                'password' => md5($passwordBaru));
            $hasil = $this->customer->UpdatePassword($data, array('customerID' =>
            $customerID));
            if($hasil == 1){

```

```

$this->session->set_flashdata("suksesPass", "<div class='alert alert-
success'><strong>Update Password BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'profile');}else{
$this->session->set_flashdata("alertPass", "<div class='alert alert-
danger'><strong>Update Password GAGAL di lakukan</strong></div>");
header('location:'.base_url().'profile');}}else{
$customerID = $this->session->userdata('customerID');
$data = array(
'user' => $this->customer->GetCustomerByID($customerID),
'title' => "customer - Mark The Barber");
$this->session->set_flashdata("alertPass", "<div class='alert alert-
danger'><strong>Password tidak cocok! atau Password kurang dari 8
karakter</strong></div>");
header('location:'.base_url().'profile');}}else{
$customerID = $this->session->userdata('customerID');
$data = array(
'user' => $this->customer->GetCustomerByID($customerID),
'title' => "customer - Mark The Barber");
$this->session->set_flashdata("alertPass", "<div class='alert alert-
danger'><strong>Password tidak cocok! atau Password Kurang dari 8
karakter</strong></div>");
header('location:'.base_url().'profile');}}}?>

```

### Controller Reservasi

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class reservasi extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->cek_login();
$this->load->model('m_customer','customer',TRUE);
$this->load->model('m_karyawan','karyawan',TRUE);
$this->load->model('m_reservasi','reservasi',TRUE);
$this->load->model('m_treatment','treatment',TRUE);
$this->load->model('m_produk','produk',TRUE);
$this->load->model('m_gayarambut','gayarambut',TRUE);
$this->load->model('m lembur','lembur',TRUE);
$this->load->model('m_gaji','gaji',TRUE);}
private function cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'backend');}}

public function index(){
$status = $this->reservasi->CekStatus();
if($status == 1){
if(date("H:i:s") <= "08:00:00" OR date("H:i:s") >= "20:00:00"){
redirect(base_url().'informasi');}else{
$customerID = $this->session->userdata('customerID');
$data = array(
'user' => $this->customer->GetCustomerByID($customerID),
'barberman' => $this->karyawan->GetAllBarberman(),
'gayarambut' => $this->gayarambut->GetTren(),
'title' => "Reservasi - Mark The Barber");
$this->load->view('reservasi/add_reservasi', $data);}}else{
redirect(base_url().'informasi');}}

//simpan data reservasi
public function simpan(){
$kodeReservasi = $this->input->post('kodeReservasi');
$customerID = $this->input->post('customerID');
$kodeRambut = $this->input->post('kodeRambut');
$namaCustomer = $this->input->post('namaCustomer');
$tglReservasi = $this->input->post('tglReservasi');
$waktuReservasi = $this->input->post('waktuReservasi');
$data = array(
'kodeReservasi' => $kodeReservasi,
'customerID' => $customerID,
'kodeRambut' => $kodeRambut,
'namaCustomer' => $namaCustomer,
'tglReservasi' => $tglReservasi,
'waktuReservasi' => $waktuReservasi,
'status' => "Menunggu");

```

```

$this->reservasi->Simpan($data);
redirect('reservasi/addtreatment/'. $kodeReservasi, $data);}

//form tambah treatment reservasi
public function addtreatment($kodeReservasi){
$customerID = $this->session->userdata('customerID');
//ambil tgl reservasi
$tglReservasi = $this->db->select('tglReservasi')
->get_where('reservasi', array('kodeReservasi' => $kodeReservasi))
->row()
->tglReservasi;

//ambil waktu reservasi
$waktuReservasi = $this->db->select('waktuReservasi')
->get_where('reservasi', array('kodeReservasi' => $kodeReservasi))
->row()
->waktuReservasi;

$data = array(
'user' => $this->customer->GetCustomerByID($customerID),
'barber' => $this->karyawan->GetAllBarbermanAvailable($tglReservasi),
'treatment' => $this->treatment->GetAll(),
'gayarambut' => $this->gayarambut->GetAll(),
'produk' => $this->produk->GetAllProduk(),
'kodeReservasi' => $kodeReservasi,
'tglReservasi' => $tglReservasi,
'waktuReservasi' => $waktuReservasi,
'status' => $this->reservasi->CekStatus(),
'title' => "Reservasi - Mark The Barber");
$this->load->view('reservasi/add_treatment', $data);}

//insert table r_treatment
public function simpantreatment(){
//POST
$kodeReservasi = $this->input->post('kodeReservasi');
//form validasi
$this->form_validation->set_error_delimiters('<div class="input-required">',
'</div>');
$this->form_validation->set_rules('kodeTreatment', 'Treatment', 'required');
if($this->form_validation->run() == TRUE){
//ambil tgl reservasi
$tglReservasi = $this->input->post('tglReservasi');
$waktuReservasi = $this->input->post('waktuReservasi');
$karyawanID = $this->input->post('karyawanID');
$kodeTreatment = $this->input->post('kodeTreatment');
$durasiTreatment = $this->input->post('durasi');
$kodeReservasi = $this->input->post('kodeReservasi');
$harga = $this->input->post('harga');
//cek jadwal ketersediaan berdasarkan tgl, waktu, dan barberman
$jadwal = $this->reservasi->CekJadwal($tglReservasi, $waktuReservasi,
$karyawanID);
if($jadwal->num_rows() == 0)
//jika jadwal tersedia lakukan reservasi sesuai jadwal{
//cek apakah ada barberman yang lembur
$hasil = $this->lembur->GetBarbermanLembur($karyawanID, $tglReservasi);
if($hasil->num_rows() == 1){
//ambil gaji lembur
$lembur = $this->db->select('lembur')
->get_where('gaji_barberman', array('karyawanID' => $karyawanID,
'kodeTreatment' => $kodeTreatment))
->row()
->lembur;
$data['kodeReservasi'] = $kodeReservasi;
$data['karyawanID'] = $karyawanID;
$data['kodeTreatment'] = $kodeTreatment;
$data['harga'] = $harga;
$data['durasi'] = $durasiTreatment;
$data['sharing'] = $lembur;
$data['status'] = "Lembur";
$this->reservasi->SimpanTreatment($data);}else{
//ambil gaji regular
$regular = $this->db->select('regular')
->get_where('gaji_barberman', array('karyawanID' => $karyawanID,
'kodeTreatment' => $kodeTreatment))

```

```

->row()
->regular;
$data['kodeReservasi'] = $kodeReservasi;
$data['karyawanID'] = $karyawanID;
$data['kodeTreatment'] = $kodeTreatment;
$data['harga'] = $harga;
$data['durasi'] = $durasiTreatment;
$data['sharing'] = $regular;
$data['status'] = "Regular";
$this->reservasi->SimpanTreatment($data);
$jam = 60;
//$time = date('H:i:s', mktime($jam));
//$jam_reservasi = date('H:i:s', $jam);
$fields = array(
'tglReservasi' => $tglReservasi,
'waktuReservasi' => $waktuReservasi,
'durasi' => $jam,
'sisaWaktu' => $jam - $durasiTreatment);
$this->reservasi->SimpanSisaWaktu($fields);
redirect('reservasi/sukses/'. $kodeReservasi);}else
//jika jadwal tidak tersedia, tampilkan pesan{
//cek sisa waktu
$sisaWaktu = $this->db->select('sisaWaktu')
->get_where('sisa_waktu', array('tglReservasi' => $tglReservasi,
'waktuReservasi' => $waktuReservasi))
->row()
->sisaWaktu;
if($sisaWaktu > $durasiTreatment){
//cek apakah ada barberman yang lembur
$hasil = $this->lembur->GetBarbermanLembur($karyawanID, $tglReservasi);
if($hasil->num_rows() == 1){
//ambil gaji lembur
$lembur = $this->db->select('lembur')
->get_where('gaji_barberman', array('karyawanID' => $karyawanID,
'kodeTreatment' => $kodeTreatment))
->row()
->lembur;
$data['kodeReservasi'] = $kodeReservasi;
$data['karyawanID'] = $karyawanID;
$data['kodeTreatment'] = $kodeTreatment;
$data['harga'] = $harga;
$data['durasi'] = $durasiTreatment;
$data['sharing'] = $lembur;
$data['status'] = "Lembur";
$this->reservasi->SimpanTreatment($data);}else{
//ambil gaji regular
$regular = $this->db->select('regular')
->get_where('gaji_barberman', array('karyawanID' => $karyawanID,
'kodeTreatment' => $kodeTreatment))
->row()
->regular;
$data['kodeReservasi'] = $kodeReservasi;
$data['karyawanID'] = $karyawanID;
$data['kodeTreatment'] = $kodeTreatment;
$data['harga'] = $harga;
$data['durasi'] = $durasiTreatment;
$data['sharing'] = $regular;
$data['status'] = "Regular";
$this->reservasi->SimpanTreatment($data);}
//id sisa waktu
$id = $this->db->select('id')
->get_where('sisa_waktu', array('tglReservasi' => $tglReservasi,
'waktuReservasi' => $waktuReservasi))
->row()
->id;
//ambil waktu reservasi
$waktu_reservasi = $this->db->select('waktuReservasi')
->get_where('reservasi', array('tglReservasi' => $tglReservasi,
'waktuReservasi' => $waktuReservasi))
->row()
->waktuReservasi;
//ambil durasi treatment
$durasi = $this->db->select('durasi')
->get_where('report_waktu_reservasi', array('tglReservasi' => $tglReservasi,

```



```

'waktuReservasi' => $waktuReservasi, 'karyawanID' => $karyawanID))
->row()
->durasi;
$jam = date_create($waktu_reservasi);
date_add($jam, date_interval_create_from_date_string($durasi.'minutes'));
$fields = array(
'id' => $id,
'tglReservasi' => $tglReservasi,
'waktuReservasi' => $waktuReservasi,
'sisaWaktu' => $sisawaktu-$durasiTreatment);
$atribut = array(
'kodeReservasi' => $kodeReservasi,
'waktuReservasi' => date_format($jam, 'H:i:s'));
$this->reservasi->UpdateSisaWaktu($fields);
$this->reservasi->UpdateWaktuReservasi($atribut);
redirect('reservasi/sukses/'.$kodeReservasi);}else{
//pesan tidak tersedia
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Maaf, jadwal dengan Barberman yang Anda pilih tidak tersedia atau
sudah dipesan!</strong></div>");
header('location:'.base_url().'reservasi/addtreatment/'.$kodeReservasi);}else{
//pesan tidak tersedia
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Mohon untuk mengisi data dengan lengkap!</strong></div>");
header('location:'.base_url().'reservasi/addtreatment/'.$kodeReservasi);}

public function sukses($kodeReservasi){
$customerID = $this->session->userdata('customerID');
//ambil data customer
$data = array(
'user' => $this->customer->GetCustomerByID($customerID),
'status' => $this->reservasi->CekStatus(),
'reservasi' => $this->reservasi->GetReservasiByKode($kodeReservasi),
'title' => "Reservasi Sukses - Mark The Barber");
$this->load->view('reservasi/sukses_reservasi', $data);}

function batal($kodeReservasi){
$hasil = $this->reservasi->Batal(array('kodeReservasi' => $kodeReservasi));
if($hasil == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Reservasi dibatalkan!</strong></div>");
header('location:'.base_url().'reservasi');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>GAGAL membatalkan reservasi!</strong></div>");
header('location:'.base_url().'reservasi');}}

public function informasi(){
$customerID = $this->session->userdata('customerID');
//ambil data customer
$data = array(
'user' => $this->customer->GetCustomerByID($customerID),
'status' => $this->reservasi->CekStatus(),
'title' => "Reservasi Tutup - Mark The Barber");
$this->load->view('reservasi/tutup_reservasi', $data);}?>

```

## Controller Produk

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Produk extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->cek_login();
$this->load->model('m_karyawan','karyawan',TRUE);
$this->load->model('m_produk','produk',TRUE);
$this->load->model('m_kategori_produk','kategori',TRUE);}

private function cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'office/backend');}}

public function index($offset=0){
$karyawanID = $this->session->userdata('karyawanID');

```

```

$jumlah = $this->db->get('produk');
$config['base_url'] = base_url().'office/produk/index';
$config['total_rows'] = $jumlah->num_rows();
$config['per_page'] = 8; /*Jumlah data yang dipanggil perhalaman*/
$config['uri_segment'] = 4; /*data selanjutnya di parse diurisegmen 3*/
/*Class bootstrap pagination yang digunakan*/
$config['full_tag_open'] = "<ul class='pagination pagination-sm'
style='position:relative; top:-25px;'>";
$config['full_tag_close'] = "</ul>";
$config['num_tag_open'] = '<li>';
$config['num_tag_close'] = '</li>';
$config['cur_tag_open'] = "<li class='disabled'><li class='active'><a href='#'>";
$config['cur_tag_close'] = "<span class='sr-only'></span></a></li>";
$config['next_tag_open'] = "<li>";
$config['next_tagl_close'] = "</li>";
$config['prev_tag_open'] = "<li>";
$config['prev_tagl_close'] = "</li>";
$config['first_tag_open'] = "<li>";
$config['first_tagl_close'] = "</li>";
$config['last_tag_open'] = "<li>";
$config['last_tagl_close'] = "</li>";
$this->pagination->initialize($config);
$num = $config['per_page'];
//ambil data karyawan
$data = array(
    'halaman' => $this->pagination->create_links(),
    'offset' => $offset,
    'user' => $this->karyawan->GetByID($karyawanID),
    'produk' => $this->produk->GetAll($num, $offset),
    'title' => "Produk - Mark The Barber";
    $this->load->view('office/produk/data_produk', $data);

function addproduk(){
    $karyawanID = $this->session->userdata('karyawanID');
    //ambil data
    $data = array(
        'user' => $this->karyawan->GetByID($karyawanID),
        'kategori' => $this->kategori->GetAll(),
        'title' => "Produk - Mark The Barber";
        $this->load->view('office/produk/add_produk', $data);

function editproduk($kodeProduk = 0){
    $karyawanID = $this->session->userdata('karyawanID');
    $data_produk = $this->produk->GetProduk("where kodeProduk = '$kodeProduk'")-
    >result_array();
    /*menjadikan kategori ke array*/
    $kategori_post_array = array();
    foreach($this->produk->GetProduk("where kodeProduk = '$kodeProduk'")-
    >result_array() as $kat){
        $kategori_post_array[] = $kat['katID'];
    }
    $data = array(
        'user' => $this->karyawan->GetByID($karyawanID),
        'kode' => $data_produk[0]['kodeProduk'],
        'nama' => $data_produk[0]['namaProduk'],
        'harga' => $data_produk[0]['harga'],
        'desc' => $data_produk[0]['keterangan'],
        'foto' => $data_produk[0]['gambar'],
        'kategori' => $this->kategori->GetAll(),
        'kat_post' => $kategori_post_array,
        'title' => "Produk - Mark The Barber";
        $this->load->view('office/produk/edit_produk', $data);

public function simpan(){
    //form validasi
    $this->form_validation->set_error_delimiters('<div class="input-required">',
    '</div>');
    $this->form_validation->set_rules('katID', 'Kategori ID', 'required');
    $this->form_validation->set_rules('namaProduk', 'Nama Produk', 'required');
    $this->form_validation->set_rules('harga', 'Harga', 'required|numeric');
    if($this->form_validation->run() == TRUE){
        $config = array(
            'upload_path' => './upload/produk',
            'allowed_types' => 'gif|jpg|JPG|jpeg|png',
            'max_size' => '2048',);

```

```

$this->load->library('upload', $config);
$this->upload->do_upload('file_upload');
$upload_data = $this->upload->data();
$kodeProduk = $_POST['kodeProduk'];
$katID = $_POST['katID'];
$namaProduk = $_POST['namaProduk'];
$harga = $_POST['harga'];
$keterangan = $_POST['keterangan'];
$file_name = $upload_data['file_name'];
$data = array(
'kodeProduk'=> $kodeProduk,
'katID'=> $katID,
'namaProduk' => $namaProduk,
'harga' => $harga,
'keterangan' => $keterangan,
'gambar' => $file_name,);
$result = $this->produk->Simpan('produk', $data);
if($result == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Simpan data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/produk');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Simpan data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/produk');} }else{
$this->addproduk();} }

function update(){
if($_FILES['file_upload']['error'] == 0):
$config = array(
'upload_path' => './upload/produk',
'allowed_types' => 'gif|jpg|JPG|png',
'max_size' => '2048',);
$this->load->library('upload', $config);
$this->upload->do_upload('file_upload');
$upload_data = $this->upload->data();
$file_name = $upload_data['file_name'];
else:
$file_name = $this->input->post('gambar');
endif;
$data = array(
'kodeProduk' => $this->input->post('kodeProduk'),
'katID' => $this->input->post('katID'),
'namaProduk' => $this->input->post('namaProduk'),
'harga' => $this->input->post('harga'),
'keterangan' => $this->input->post('keterangan'),
'gambar' => $file_name,);
$res = $this->produk->Updateproduk($data);
if($res>=0){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Update data BERHASIL di lakukan</strong></div>");
header('location:'.base_url().'office/produk');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Update data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/produk');} }

function hapus($kodeProduk){
$result = $this->produk->Hapus('produk', array('kodeProduk' => $kodeProduk));
if($result == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Hapus data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/produk');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Hapus data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/produk');} }?>

```

## Controller Treatment

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Treatment extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->cek_login();

```

```

$this->load->model('m_karyawan','karyawan',TRUE);
$this->load->model('m_treatment','treatment',TRUE);

private function _cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'office/backend');}}

public function index(){
$karyawanID = $this->session->userdata('karyawanID');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'treatment' => $this->treatment->GetAll(),
'title' => "Treatment - Mark The Barber");
$this->load->view('office/treatment/data_treatment', $data);}

function tambahtreatment(){
$karyawanID = $this->session->userdata('karyawanID');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'title' => "Treatment - Mark The Barber");
$this->load->view('office/treatment/add_treatment', $data);}

function ubahtreatment($kodeTreatment){
$karyawanID = $this->session->userdata('karyawanID');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'treat' => $this->treatment->GetByKode($kodeTreatment),
'title' => "Treatment - Mark The Barber");
$this->load->view('office/treatment/edit_treatment', $data);}

public function simpan(){
//form validasi
$this->form_validation->set_error_delimiters('<div class="input-required">',
'</div>');
$this->form_validation->set_rules('namaTreatment', 'Nama Treatment', 'required');
$this->form_validation->set_rules('harga', 'Harga', 'required|numeric');
$this->form_validation->set_rules('durasi', 'Durasi', 'required|numeric');

if($this->form_validation->run() == TRUE){
$config = array(
'upload_path' => './upload/treatment',
'allowed_types' => 'gif|jpg|JPG|JPEG|png',
'max_size' => '2048',);
$this->load->library('upload', $config);
$this->upload->do_upload('file_upload');
$upload_data = $this->upload->data();
$kodeTreatment = $_POST['kodeTreatment'];
$namaTreatment = $_POST['namaTreatment'];
$harga = $_POST['harga'];
$durasi = $_POST['durasi'];
$keterangan = $_POST['keterangan'];
$file_name = $upload_data['file_name'];
$data = array(
'kodeTreatment'=> $kodeTreatment,
'namaTreatment' => $namaTreatment,
'harga' => $harga,
'durasi' => $durasi,
'keterangan' => $keterangan,
'gambar' => $file_name,);
$result = $this->treatment->Simpan($data);
if($result == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Simpan data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/treatment');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Simpan data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/treatment');}}else{
$this->tambahtreatment();}
}

function update(){
//form validasi

```

```

$this->form_validation->set_error_delimiters('<div class="input-required">',
'</div>');
$this->form_validation->set_rules('namaTreatment', 'Nama Treatment', 'required');
$this->form_validation->set_rules('harga', 'Harga', 'required|numeric');
$this->form_validation->set_rules('durasi', 'Durasi', 'required|numeric');
if($this->form_validation->run() == TRUE){
if($_FILES['file_upload']['error'] == 0):
$config = array(
'upload_path' => './upload/treatment',
'allowed_types' => 'gif|jpg|JPG|jpeg|png',
'max_size' => '2048',);
$this->load->library('upload', $config);
$this->upload->do_upload('file_upload');
$upload_data = $this->upload->data();
$file_name = $upload_data['file_name'];
else:
$file_name = $this->input->post('gambar');
endif;
$kodeTreatment = $this->input->post('kodeTreatment');
$data = array(
'namaTreatment' => $this->input->post('namaTreatment'),
'harga' => $this->input->post('harga'),
'durasi' => $this->input->post('durasi'),
'keterangan' => $this->input->post('keterangan'),
'gambar' => $file_name,);
$hasil = $this->treatment->Update($data, array('kodeTreatment' =>
$kodeTreatment));
if($hasil == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Update data BERHASIL di lakukan</strong></div>");
header('location:'.base_url().'office/treatment');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Update data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/treatment');}}else{
$kodeTreatment = $this->input->post('kodeTreatment');
$this->ubahtreatment($kodeTreatment);}

function hapus($kodeTreatment){
$result = $this->treatment->Hapus('treatment', array('kodeTreatment' =>
$kodeTreatment));
if($result == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Hapus data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/treatment');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Hapus data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/treatment');}}}?>

```

### Controller Gaya Rambut

```

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class Gayarambut extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->cek_login();
$this->load->model('m_karyawan', 'karyawan', TRUE);
$this->load->model('m_gayarambut', 'gayarambut', TRUE);}
private function _cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'office/backend');}}

public function index(){
$karyawanID = $this->session->userdata('karyawanID');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'gayarambut' => $this->gayarambut->GetAll(),
'title' => "Gaya Rambut - Mark The Barber");
$this->load->view('office/gayarambut/data_gayarambut', $data);}

function addgayarambut(){
$karyawanID = $this->session->userdata('karyawanID');

```

```

//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'title' => "Gaya Rambut - Mark The Barber");
$this->load->view('office/gayarambut/add_gayarambut', $data);}

function ubahgayarambut($kodeRambut){
$karyawanID = $this->session->userdata('karyawanID');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'gayarambut' => $this->gayarambut->GetByKode($kodeRambut),
'title' => "Gaya Rambut - Mark The Barber");
$this->load->view('office/gayarambut/edit_gayarambut', $data);}

public function simpan(){
//form validasi
$this->form_validation->set_error_delimiters('<div class="input-required">',
'</div>');
$this->form_validation->set_rules('namaRambut', 'Nama Rambut', 'required');
if($this->form_validation->run() == TRUE){
$config = array(
'upload_path' => './upload/gayarambut',
'allowed_types' => 'gif|jpg|JPG|jpeg|png',
'max_size' => '2048',);
$this->load->library('upload', $config);
$this->upload->initialize($config);
//upload gambar depan
$this->upload->do_upload('file_upload');
$upload_data = $this->upload->data();
//upload gambar samping
$this->upload->do_upload('file_upload_samping');
$upload_data_samping = $this->upload->data();
//upload gambar belakang
$this->upload->do_upload('file_upload_belakang');
$upload_data_belakang = $this->upload->data();
$kodeRambut = $_POST['kodeRambut'];
$namaRambut = $_POST['namaRambut'];
$file_name = $upload_data['file_name'];
$file_name_samping = $upload_data_samping['file_name'];
$file_name_belakang = $upload_data_belakang['file_name'];
$data = array(
'kodeRambut'=> $kodeRambut,
'namaRambut' => $namaRambut,
'gambarDepan' => $file_name,
'gambarSamping' => $file_name_samping,
'gambarBelakang' => $file_name_belakang,);
$result = $this->gayarambut->Simpan($data);
if($result == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Simpan data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/gayarambut');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Simpan data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/gayarambut');}}else{
$this->addgayarambut();}}

function update(){
if($_FILES['file_upload']['error'] == 0):
$config = array(
'upload_path' => './upload/gayarambut',
'allowed_types' => 'gif|jpg|JPG|jpeg|JPEG|png',
'max_size' => '2048',
'overwrite' => TRUE,);
$this->load->library('upload', $config);
$this->upload->initialize($config);
//upload gambar depan
$this->upload->do_upload('file_upload');
$upload_data = $this->upload->data();
//upload gambar samping
$this->upload->do_upload('file_upload_samping');
$upload_data_samping = $this->upload->data();
//upload gambar belakang
$this->upload->do_upload('file_upload_belakang');

```

```

$upload_data_belakang = $this->upload->data();
$file_name = $upload_data['file_name'];
$file_name_samping = $upload_data_samping['file_name'];
$file_name_belakang = $upload_data_belakang['file_name'];
$data = array(
    'namaRambut' => $this->input->post('namaRambut'),
    'gambarDepan' => $file_name,
    'gambarSamping' => $file_name_samping,
    'gambarBelakang' => $file_name_belakang,);
else;
$file_name = $this->input->post('gambarDepan');
$file_name_samping = $this->input->post('gambarSamping');
$file_name_belakang = $this->input->post('gambarBelakang');
endif;
$data = array(
    'namaRambut' => $this->input->post('namaRambut'),
    'gambarDepan' => $file_name,
    'gambarSamping' => $file_name_samping,
    'gambarBelakang' => $file_name_belakang,);
$kodeRambut = $this->input->post('kodeRambut');
$res = $this->gayarambut->Update($data, array('kodeRambut' => $kodeRambut));
if($res>=0){
    $this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Update data BERHASIL di lakukan</strong></div>");
header('location:'.base_url().'office/gayarambut');}else{
    $this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Update data GAGAL di
lakukan</strong></div>");header('location:'.base_url().'office/gayarambut');}}

function hapus($kodeRambut){
    $result = $this->gayarambut->Hapus(array('kodeRambut' => $kodeRambut));
    if($result == 1){
        $this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Hapus data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/gayarambut');}else{
    $this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Hapus data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/gayarambut');}}?>

```

## Controller Lembur

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Lembur extends CI_Controller{
    function __construct(){
        parent::__construct();
        date_default_timezone_set('Asia/Jakarta');
        $this->cek_login();
        $this->load->model('m_karyawan', 'karyawan', TRUE);
        $this->load->model('m_lembur', 'lembur', TRUE);
        private function cek_login(){
            if(!$this->session->userdata('logged_in')){
                redirect(base_url().'office/backend');}
        }

        public function index(){
            $karyawanID = $this->session->userdata('karyawanID');
            $data = array(
                'user' => $this->karyawan->GetByID($karyawanID),
                'lembur' => $this->lembur->GetlemburByMonth(),
                'barber' => $this->karyawan->GetAllBarberman(),
                'title' => "Lembur - Mark The Barber");
            $this->load->view('office/lembur/data_lembur', $data);}

        public function simpan(){
            //form validasi
            $this->form_validation->set_error_delimiters('<div class="input-required">',
                '</div>');
            $this->form_validation->set_rules('karyawanID', 'ID', 'required');
            $this->form_validation->set_rules('tglLembur', 'Tgl Lembur', 'required');
            if($this->form_validation->run() == TRUE){
                $karyawanID = $this->input->post('karyawanID');
                $tglLembur = $this->input->post('tglLembur');
                $data = array(
                    'karyawanID' => $karyawanID,

```

```

'tglLembur' => $tglLembur);
$hasil = $this->lembur->Simpan($data);
if($hasil == 1){
//pesan berhasil
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Simpan data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/lembur');}else{
//pesan gagal
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Simpan data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/lembur');}else{
$this->index();}

```

```

function hapus($lemburID){
$hasil = $this->lembur->Hapus(array('lemburID' => $lemburID));
if($hasil == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Hapus data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/lembur');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Hapus data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/lembur');}]}?>

```

## Controller Libur

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Libur extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->cek_login();
$this->load->model('m_karyawan','karyawan',TRUE);
$this->load->model('m_libur','libur',TRUE);}
private function cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'office/backend');}}

public function index(){
$karyawanID = $this->session->userdata('karyawanID');
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'libur' => $this->libur->GetLiburByMonth(),
'barber' => $this->karyawan->GetAllBarberman(),
'title' => "Libur - Mark The Barber");
$this->load->view('office/libur/data_libur', $data);}

public function simpan(){
//form validasi
$this->form_validation->set_error_delimiters('<div class="input-required">',
'</div>');
$this->form_validation->set_rules('karyawanID', 'ID', 'required');
$this->form_validation->set_rules('tglLibur', 'Tgl Libur', 'required');
if($this->form_validation->run() == TRUE){
$karyawanID = $this->input->post('karyawanID');
$tglLibur = $this->input->post('tglLibur');
$data = array(
'karyawanID' => $karyawanID,
'tglLibur' => $tglLibur
);
$hasil = $this->libur->Simpan($data);
if($hasil == 1){
//pesan berhasil
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Simpan data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/libur');}else{
//pesan gagal
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Simpan data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/libur');}else{
$this->index();}

function hapus($liburID){
$hasil = $this->libur->Hapus(array('liburID' => $liburID));

```



```

if($hasil == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Hapus data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/libur');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Hapus data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/libur');}}}?>

```

## Controller Gaji

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
class Gaji extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->cek_login();
$this->load->model('m_karyawan','karyawan',TRUE);
$this->load->model('m_gaji','gaji',TRUE);
$this->load->model('m_treatment','treatment',TRUE);
private function cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'office/backend');}}

public function index(){
$karyawanID = $this->session->userdata('karyawanID');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'gaji' => $this->gaji->GetAll(),
'barber' => $this->karyawan->GetAllBarberman(),
'treat' => $this->treatment->GetAll(),
'title' => "Gaji - Mark The Barber");
$this->load->view('office/gaji/data_gaji', $data);}

public function simpan(){
//form validasi
$this->form_validation->set_error_delimiters('<div class="input-required">',
'</div>');
$this->form_validation->set_rules('karyawanID', 'ID', 'required');
$this->form_validation->set_rules('kodeTreatment', 'Treatment', 'required');
$this->form_validation->set_rules('regular', 'Regular', 'required|numeric');
$this->form_validation->set_rules('lembur', 'Lembur', 'required|numeric');
if($this->form_validation->run() == TRUE){
$karyawanID = $this->input->post('karyawanID');
$kodeTreatment = $this->input->post('kodeTreatment');
$regular = $this->input->post('regular');
$lembur = $this->input->post('lembur');
$data = array(
'karyawanID' => $karyawanID,
'kodeTreatment' => $kodeTreatment,
'regular' => $regular,
'lembur' => $lembur,
'tglBuat' => date('Y-m-d'));
$hasil = $this->gaji->Simpan($data);
if($hasil == 1){
//pesan berhasil
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Simpan data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/gaji');}else{
//pesan gagal
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Simpan data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/gaji');}}else{
$this->index();}}

public function ubah($gajiID){
$karyawanID = $this->session->userdata('karyawanID');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'gaji' => $this->gaji->GetAll(),
'gajiedit' => $this->gaji->GetByID($gajiID),
'title' => "Gaji - Mark The Barber");

```

```

$this->load->view('office/gaji/edit_gaji', $data);}

public function update(){
//form validasi
$this->form_validation->set_error_delimiters('<div class="input-required">',
'</div>');
$this->form_validation->set_rules('gajiID', 'ID', 'required');
$this->form_validation->set_rules('regular', 'Regular', 'required|numeric');
$this->form_validation->set_rules('lembur', 'Lembur', 'required|numeric');
if($this->form_validation->run() == TRUE){
$gajiID = $this->input->post('gajiID');
$regular = $this->input->post('regular');
$lembur = $this->input->post('lembur');
$data = array(
'regular' => $regular,
'lembur' => $lembur,
'tglUbah' => date('Y-m-d'));
$hasil = $this->gaji->Update($data, array('gajiID' => $gajiID));
if($hasil == 1){
//pesan berhasil
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Update data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/gaji');}else{
//pesan gagal
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Update data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/gaji');}else{
$gajiID = $this->input->post('gajiID');
$this->ubah($gajiID);}

function hapus($gajiID){
$hasil = $this->gaji->Hapus(array('gajiID' => $gajiID));
if($hasil == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Hapus data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/gaji');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Hapus data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/gaji');}}}?>

```

### Controller Penggajian

```

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class Penggajian extends CI_Controller {
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->cek_login();
$this->load->model('m_karyawan', 'karyawan', TRUE);
$this->load->model('m_reservasi', 'reservasi', TRUE);
$this->load->model('m_penggajian', 'penggajian', TRUE);
private function cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'office/backend');}}

public function index(){
$karyawanID = $this->session->userdata('karyawanID');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'barber' => $this->karyawan->GetAllBarberman(),
'tahun' => date('Y'),
'title' => "Gajian Barberman - Mark The Barber");
$this->load->view('office/penggajian/form_gajian', $data);}

function tampilkan(){
if($this->input->get('informasi')){
$karyawanID = $this->session->userdata('karyawanID');
$kryID = $this->input->get('karyawanID');
$bulan = $this->input->get('bulan');
//ambil data
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),

```

```

'kry' => $this->karyawan->GetByID($kryID),
'gajian' => $this->penggajian->GetGajian($kryID, $bulan),
'totalgaji' => $this->penggajian->GetTotalGaji($kryID, $bulan),
'tanggal' => date('Y-m-d'),
'title' => "Gajian Barberman - Mark The Barber";
$this->load->view('office/penggajian/informasi_gaji', $data);}else{
$kryID = $this->input->get('karyawanID');
$bulan = $this->input->get('bulan');
//ambil data
$data = array(
'kry' => $this->karyawan->GetByID($kryID),
'gajian' => $this->penggajian->GetGajian($kryID, $bulan),
'totalgaji' => $this->penggajian->GetTotalGaji($kryID, $bulan),
'tanggal' => date('Y-m-d'),
'title' => "Gajian Barberman - Mark The Barber";
$this->load->view('office/penggajian/cetak_informasi_gaji', $data);}
public function simpan(){
$data['tglPengeluaran'] = date('Y-m-d');
$this->pengeluaran->Simpan($data);
redirect('office/pengeluaran');}
public function simpandetail(){
$pengeluaranID = $this->input->post('pengeluaranID');
$karyawanID = $this->input->post('karyawanID');
$kodeLayanan = $this->input->post('kodeLayanan');
$harga = $this->input->post('harga');
$keterangan = $this->input->post('keterangan');
$data = array(
'pengeluaranID' => $pengeluaranID,
'karyawanID' => $karyawanID,
'kodeLayanan' => $kodeLayanan,
'harga' => $harga,
'keterangan' => $keterangan);
$this->pengeluaran->SimpanDetail($data);
redirect('office/pengeluaran');}

function hapus($detailID){
$hasil = $this->pengeluaran->HapusDetail(array('detailID' => $detailID));
if($hasil == 1){
$this->session->set_flashdata("sukses", "<div class='alert alert-
success'><strong>Hapus data BERHASIL dilakukan</strong></div>");
header('location:'.base_url().'office/pengeluaran');}else{
$this->session->set_flashdata("alert", "<div class='alert alert-
danger'><strong>Hapus data GAGAL di lakukan</strong></div>");
header('location:'.base_url().'office/pengeluaran');}}}?>

```

### Controller Laporan

```

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class Laporan extends CI_Controller{
function __construct(){
parent::__construct();
date_default_timezone_set('Asia/Jakarta');
$this->load->helper('fungsidate');
$this->_cek_login();
$this->load->model('m_customer','customer',TRUE);
$this->load->model('m_karyawan','karyawan',TRUE);
$this->load->model('m_reservasi','reservasi',TRUE);
$this->load->model('m_treatment','treatment',TRUE);
$this->load->model('m_produk','produk',TRUE);
$this->load->model('m_gayarambut','gayarambut',TRUE);
$this->load->model('m lembur','lembur',TRUE);
$this->load->model('m_gaji','gaji',TRUE);
$this->load->model('m_laporan','laporan',TRUE);}
private function _cek_login(){
if(!$this->session->userdata('logged_in')){
redirect(base_url().'office/backend');}}

//cari laporan
public function index(){
$karyawanID = $this->session->userdata('karyawanID');
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'dropdaystart' => $this->dob_dropdown->buildDayDropdown('tanggalstart'),

```

```

'dropmonthstart' => $this->dob_dropdown->buildMonthDropdown('bulanstart'),
'dropyearstart' => $this->dob_dropdown->buildYearDropdown('tahunstart'),
'dropdayend' => $this->dob_dropdown->buildDayDropdown('tanggalend'),
'dropmonthend' => $this->dob_dropdown->buildMonthDropdown('bulanend'),
'dropyearend' => $this->dob_dropdown->buildYearDropdown('tahunend'),
'dropmonth' => $this->dob_dropdown->buildMonthDropdown('bulan'),
'dropyear' => $this->dob_dropdown->buildYearDropdown('tahun'),
'title' => "Laporan - Mark The Barber";
$this->load->view('office/laporan/cari_laporan', $data);}

public function tampilkan(){
if ($this->input->get('tampilkan')){
$karyawanID = $this->session->userdata('karyawanID');
$periode = $this->input->get('periode');
if($periode == "perioderinci"){
$tanggal = $this->input->get('tanggalstart');
$bulan = $this->input->get('bulanstart');
$tahun = $this->input->get('tahunstart');
$tanggalend = $this->input->get('tanggalend');
$bulanend = $this->input->get('bulanend');
$tahunend = $this->input->get('tahunend');
$tglStart = $tahun.'-'. $bulan.'-'. $tanggal;
$tglEnd = $tahunend.'-'. $bulanend.'-'. $tanggalend;
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'treatment' => $this->laporan->GetTreatmentByPeriode($tglStart, $tglEnd),
'sumtreatment' => $this->laporan->GetTotalTreatmentByPeriode($tglStart, $tglEnd),
'produk' => $this->laporan->GetProdukByPeriode($tglStart, $tglEnd),
'sumproduk' => $this->laporan->GetTotalProdukByPeriode($tglStart, $tglEnd),
'pengeluaran' => $this->laporan->GetPengeluaranByPeriode($tglStart, $tglEnd),
'sumpengeluaran' => $this->laporan->GetTotalPengeluaranByPeriode($tglStart,
$tglEnd),
'dropdaystart' => $this->dob_dropdown->buildDayDropdown('tanggalstart'),
'dropmonthstart' => $this->dob_dropdown->buildMonthDropdown('bulanstart'),
'dropyearstart' => $this->dob_dropdown->buildYearDropdown('tahunstart'),
'dropdayend' => $this->dob_dropdown->buildDayDropdown('tanggalend'),
'dropmonthend' => $this->dob_dropdown->buildMonthDropdown('bulanend'),
'dropyearend' => $this->dob_dropdown->buildYearDropdown('tahunend'),
'dropmonth' => $this->dob_dropdown->buildMonthDropdown('bulan'),
'dropyear' => $this->dob_dropdown->buildYearDropdown('tahun'),
'tanggal' => $tglStart,
'end' => $tglEnd,
'tertanggal' => tgl_indo(date('Y-m-d')),
'title' => "Laporan - Mark The Barber");}else{
$bulan = $this->input->get('bulan');
$tahun = $this->input->get('tahun');
$tanggal = $tahun.'-'. $bulan;
$data = array(
'user' => $this->karyawan->GetByID($karyawanID),
'treatment' => $this->laporan->GetTreatment($bulan, $tahun),
'sumtreatment' => $this->laporan->GetTotalTreatment($bulan, $tahun),
'produk' => $this->laporan->GetProduk($bulan, $tahun),
'sumproduk' => $this->laporan->GetTotalProduk($bulan, $tahun),
'pengeluaran' => $this->laporan->GetPengeluaran($bulan, $tahun),
'sumpengeluaran' => $this->laporan->GetTotalPengeluaran($bulan, $tahun),
'dropdaystart' => $this->dob_dropdown->buildDayDropdown('tanggalstart'),
'dropmonthstart' => $this->dob_dropdown->buildMonthDropdown('bulanstart'),
'dropyearstart' => $this->dob_dropdown->buildYearDropdown('tahunstart'),
'dropdayend' => $this->dob_dropdown->buildDayDropdown('tanggalend'),
'dropmonthend' => $this->dob_dropdown->buildMonthDropdown('bulanend'),
'dropyearend' => $this->dob_dropdown->buildYearDropdown('tahunend'),
'dropmonth' => $this->dob_dropdown->buildMonthDropdown('bulan'),
'dropyear' => $this->dob_dropdown->buildYearDropdown('tahun'),
'tanggal' => $tanggal,
'tertanggal' => tgl_indo(date('Y-m-d')),
'title' => "Laporan - Mark The Barber");}
$this->load->view('office/laporan/laporan_periode', $data);}else{
$karyawanID = $this->session->userdata('karyawanID');
$periode = $this->input->get('periode');
if($periode == "perioderinci"){
$tanggal = $this->input->get('tanggalstart');
$bulan = $this->input->get('bulanstart');
$tahun = $this->input->get('tahunstart');
$tanggalend = $this->input->get('tanggalend');

```

```

$bulanend = $this->input->get('bulanend');
$tahunend = $this->input->get('tahunend');
$tglStart = $tahun.'-'. $bulan.'-'. $tanggal;
$tglEnd = $tahunend.'-'. $bulanend.'-'. $tanggalend;
$data = array(
    'user' => $this->karyawan->GetByID($karyawanID),
    'treatment' => $this->laporan->GetTreatmentByPeriode($tglStart, $tglEnd),
    'sumtreatment' => $this->laporan->GetTotalTreatmentByPeriode($tglStart, $tglEnd),
    'produk' => $this->laporan->GetProdukByPeriode($tglStart, $tglEnd),
    'sumproduk' => $this->laporan->GetTotalProdukByPeriode($tglStart, $tglEnd),
    'pengeluaran' => $this->laporan->GetPengeluaranByPeriode($tglStart, $tglEnd),
    'sumpengeluaran' => $this->laporan->GetTotalPengeluaranByPeriode($tglStart,
    $tglEnd),
    'tanggal' => $tglStart,
    'end' => $tglEnd,
    'tertanggal' => tgl_indo(date('Y-m-d')),
    'title' => "Laporan - Mark The Barber");}else{
$bulan = $this->input->get('bulan');
$tahun = $this->input->get('tahun');
$tanggal = $tahun.'-'. $bulan;
$data = array(
    'user' => $this->karyawan->GetByID($karyawanID),
    'treatment' => $this->laporan->GetTreatment($bulan, $tahun),
    'sumtreatment' => $this->laporan->GetTotalTreatment($bulan, $tahun),
    'produk' => $this->laporan->GetProduk($bulan, $tahun),
    'sumproduk' => $this->laporan->GetTotalProduk($bulan, $tahun),
    'pengeluaran' => $this->laporan->GetPengeluaran($bulan, $tahun),
    'sumpengeluaran' => $this->laporan->GetTotalPengeluaran($bulan, $tahun),
    'tanggal' => $tanggal,
    'tertanggal' => tgl_indo(date('Y-m-d')),
    'title' => "Laporan - Mark The Barber");}
$this->load->view('office/laporan/cetak_laporan', $data);}}?>

```

## Model Autentikasi

```

<?php if (! defined('BASEPATH')) exit('No direct script access allowed');
class M_autentikasi extends CI_Model {
    public function __construct() {
        parent::__construct();
        $this->load->library('session');}

    public function login_mark($karyawanID, $password) {
        $this->db->select('karyawanID, password');
        $this->db->from('karyawan');
        $this->db->where('karyawanID', $karyawanID);
        $this->db->where('password', $password);
        $this->db->limit(1);
        $query = $this->db->get();
        if($query->num_rows() == 1) {
            return TRUE;}else {
            return false;}}

    public function login_member($customerID, $email, $password) {
        $this->db->select('*');
        $this->db->from('customer');
        $this->db->where('customerID', $customerID);
        $this->db->where('email', $email);
        $this->db->where('password', $password);
        $this->db->where('aktif', 1);
        $this->db->limit(1);

        $query = $this->db->get();

        if($query->num_rows() == 1) {
            return TRUE;}else {
            return false;}}?>

```

## Model Customer

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_customer extends CI_Model{
    protected $table = 'customer';

```

```

function __construct(){
parent::__construct();}
//ambil data customer
function GetCustomer($data) {
$query = $this->db->get_where('customer', $data);
return $query;}
//cek data customer berdasarkan email
function GetCustomerDesc() {
$query = $this->db->query("SELECT * FROM customer order by customerID desc limit
1");
return $query;}

//cek data customer berdasarkan email
function CekEmail($email) {
$query = $this->db->query("SELECT * FROM customer WHERE email = '$email'");
return $query;}

public function GetCustomerByID($customerID){
$query = $this->db->query("SELECT * FROM customer WHERE customerID =
'$customerID'");
$hasil = $query->row_array();
return $hasil;}

public function GetDetailRiwayat($kodeReservasi){
$query = $this->db->query("SELECT k.namaDepan, t.namaTreatment FROM r_treatment
rt, karyawan k, treatment t WHERE rt.kodeReservasi = '$kodeReservasi' AND
rt.karyawanID = k.karyawanID AND rt.kodeTreatment = t.kodeTreatment");
$hasil = $query->result_array();
return $hasil;}

public function GetRiwayatReservasi($customerID){
$query = $this->db->query("SELECT r.kodeReservasi, r.tglReservasi,
r.waktuReservasi, k.namaDepan, r.status FROM reservasi r, r_treatment rt, karyawan
k WHERE r.customerID = $customerID AND rt.kodeReservasi = r.kodeReservasi AND
rt.karyawanID = k.karyawanID GROUP BY r.kodeReservasi");
$hasil = $query->result_array();
return $hasil;}

public function GetByPass($customerID, $passwordLama){
$query = $this->db->query("SELECT * FROM customer WHERE customerID = '$customerID'
AND password = '$passwordLama'");
return $query;}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}

public function UpdatePassword($data,$where){
return $this->db->update($this->table, $data, $where);}}?>

```

## Model Gaji

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_gaji extends CI_Model{
protected $table = 'gaji_barberman';
function __construct(){
parent::__construct();}

public function GetAll(){
$query = $this->db->query("SELECT g.*, k.namaDepan, t.namaTreatment FROM
gaji_barberman g, karyawan k, treatment t WHERE g.karyawanID = k.karyawanID AND
g.kodeTreatment = t.kodeTreatment ORDER BY g.karyawanID ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetByID($gajiID){
$query = $this->db->query("SELECT * FROM gaji_barberman WHERE gajiID = $gajiID");
$hasil = $query->row_array();
return $hasil;}
//ambil tarif gaji_barberman lembur
public function GetSharingLembur($karyawanID, $tCode){
$query = $this->db->query("SELECT s_overtime FROM gaji_barberman WHERE karyawanID
= $karyawanID AND kodeTreatment = '$tCode'");
$query->result();}

```

```

//ambil tarif gaji barberman regular
public function GetSharingRegular($karyawanID, $tCode){
$query = $this->db->query("SELECT s_regular FROM gaji_barberman WHERE karyawanID =
$karyawanID AND kodeTreatment = '$tCode'");
$query->result();}

//batas crud data
public function Simpan($data){
return $this->db->insert($this->table, $data);}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}

public function Hapus($where){
return $this->db->delete($this->table, $where);}}?>

```

## Model Gaya Rambut

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_gayarambut extends CI_Model{
protected $table = 'gaya_rambut';
function __construct(){
parent::__construct();}

public function GetAll(){
$query = $this->db->query("SELECT * FROM gaya_rambut ORDER BY kodeRambut ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetTren(){
$query = $this->db->query("SELECT * FROM view_tren_rambut ORDER BY kodeRambut ASC LIMIT 10");
$hasil = $query->result_array();
return $hasil;}

public function GetByKode($kodeRambut){
$query = $this->db->query("SELECT * FROM gaya_rambut WHERE kodeRambut =
'$kodeRambut'");
$hasil = $query->row_array();
return $hasil;}

public function Simpan($data){
return $this->db->insert($this->table, $data);}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}

public function Hapus($where){
return $this->db->delete($this->table, $where);}}?>

```

## Model Karyawan

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_karyawan extends CI_Model{
protected $table = 'karyawan';
function __construct(){
parent::__construct();}
public function GetAll(){
$query = $this->db->query("SELECT * FROM karyawan ORDER BY karyawanID ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetAllManager(){
$query = $this->db->query("SELECT * FROM karyawan WHERE level = 'Manager' ORDER BY
karyawanID ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetAllCashier(){
$query = $this->db->query("SELECT * FROM karyawan WHERE level = 'Cashier' ORDER BY
karyawanID ASC");
$hasil = $query->result_array();
return $hasil;}

```

```

return $hasil;}

public function GetAllBarberman(){
$query = $this->db->query("SELECT * FROM karyawan WHERE level = 'Barberman' ORDER
BY karyawanID ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetAllBarbermanAvailable($tglReservasi){
$query = $this->db->query("SELECT karyawanID, namaDepan FROM karyawan WHERE
level='Barberman' AND karyawanID NOT IN (SELECT k.karyawanID from karyawan k
,libur_barberman l where l.karyawanID = k.karyawanID AND l.tglLibur =
'$tglReservasi')");
$hasil = $query->result_array();
return $hasil;}

public function GetByID($karyawanID){
$query = $this->db->query("SELECT * FROM karyawan WHERE karyawanID =
'$karyawanID'");
$hasil = $query->row_array();
return $hasil;}
public function GetByPass($karyawanID, $passwordLama){
$query = $this->db->query("SELECT * FROM karyawan WHERE karyawanID = '$karyawanID'
AND password = '$passwordLama'");
return $query;}

//batas crud data
public function Simpan($data){
return $this->db->insert($this->table, $data);}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}

public function UpdatePassword($data,$where){
return $this->db->update($this->table, $data, $where);}

public function Hapus($where){
return $this->db->delete($this->table, $where);}?>

```

## Model Laporan

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_laporan extends CI_Model{
function __construct(){
parent::__construct();}

//periode bulan
public function GetTreatment($bulan, $tahun){
$query = $this->db->query("SELECT namaTreatment, sum(jumlah) as jumlah, harga,
tglReservasi, status, sum(subTotal) as subTotal FROM
report_semua_subtotal_treatment WHERE month(tglReservasi) = $bulan AND
year(tglReservasi) = $tahun AND status = 'Marked' GROUP BY namaTreatment");
$hasil = $query->result_array();
return $hasil;}

public function GetTotalTreatment($bulan, $tahun){
$query = $this->db->query("SELECT SUM(subTotal) FROM
report_semua_subtotal_treatment WHERE month(tglReservasi) = $bulan AND
year(tglReservasi) = $tahun AND status = 'Marked'");
$hasil = $query->row_array();
return $count = $hasil['SUM(subTotal)'];}

public function GetProduk($bulan, $tahun){
$query = $this->db->query("SELECT namaProduk, sum(qty) as qty, harga,
tglReservasi, status, sum(subTotal) as subTotal FROM report_semua_subtotal_produk
WHERE month(tglReservasi) = $bulan AND year(tglReservasi) = $tahun AND status =
'Marked' GROUP BY namaProduk");
$hasil = $query->result_array();
return $hasil;}

public function GetTotalProduk($bulan, $tahun){

```



```

$query = $this->db->query("SELECT SUM(subTotal) FROM report_semua_subtotal_produk
WHERE month(tglReservasi) = $bulan AND year(tglReservasi) = $tahun AND status =
'Marked'");
$hasil = $query->row_array();
return $count = $hasil['SUM(subTotal)'];}

public function GetPengeluaran($bulan, $tahun){
$query = $this->db->query("SELECT *, sum(harga) as total from report_pengeluaran
WHERE month(tglPengeluaran) = $bulan AND year(tglPengeluaran) = $tahun GROUP BY
kodeLayanan");
$hasil = $query->result_array();
return $hasil;}

public function GetTotalPengeluaran($bulan, $tahun){
$query = $this->db->query("SELECT SUM(dp.harga) FROM pengeluaran p,
detail_pengeluaran dp WHERE dp.pengeluaranID = p.pengeluaranID AND
month(p.tglPengeluaran) = $bulan AND year(p.tglPengeluaran) = $tahun");
$hasil = $query->row_array();
return $count = $hasil['SUM(dp.harga)'];}
//end of periode bulan

//periode between start end
public function GetTreatmentByPeriode($tglStart, $tglEnd){
$query = $this->db->query("SELECT * FROM report_semua_subtotal_treatment WHERE
tglReservasi BETWEEN '$tglStart' AND '$tglEnd' AND status = 'Marked' GROUP BY
namaTreatment, tglReservasi");
$hasil = $query->result_array();
return $hasil;}

public function GetTotalTreatmentByPeriode($tglStart, $tglEnd){
$query = $this->db->query("SELECT SUM(subTotal) FROM
report_semua_subtotal_treatment WHERE tglReservasi BETWEEN '$tglStart' AND
'$tglEnd' AND status = 'Marked'");
$hasil = $query->row_array();
return $count = $hasil['SUM(subTotal)'];}

public function GetProdukByPeriode($tglStart, $tglEnd){
$query = $this->db->query("SELECT * FROM report_semua_subtotal_produk WHERE
tglReservasi BETWEEN '$tglStart' AND '$tglEnd' AND status = 'Marked' GROUP BY
namaProduk, tglReservasi, status");
$hasil = $query->result_array();
return $hasil;}

public function GetTotalProdukByPeriode($tglStart, $tglEnd){
$query = $this->db->query("SELECT SUM(subTotal) FROM report_semua_subtotal_produk
WHERE tglReservasi BETWEEN '$tglStart' AND '$tglEnd' AND status = 'Marked'");
$hasil = $query->row_array();
return $count = $hasil['SUM(subTotal)'];}

public function GetPengeluaranByPeriode($tglStart, $tglEnd){
$query = $this->db->query("SELECT *, sum(harga) as total from report_pengeluaran
WHERE tglPengeluaran BETWEEN '$tglStart' AND '$tglEnd' GROUP BY kodeLayanan");
$hasil = $query->result_array();
return $hasil;}

public function GetTotalPengeluaranByPeriode($tglStart, $tglEnd){
$query = $this->db->query("SELECT SUM(dp.harga) FROM pengeluaran p,
detail_pengeluaran dp WHERE dp.pengeluaranID = p.pengeluaranID AND
p.tglPengeluaran BETWEEN '$tglStart' AND '$tglEnd'");
$hasil = $query->row_array();
return $count = $hasil['SUM(dp.harga)'];}
//end of periode between start end?>

```

## Model Lembur

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_lembur extends CI_Model{
protected $table = 'lembur_barberman';
function __construct(){
parent::__construct();}

public function GetAll(){
$query = $this->db->query("SELECT * FROM lembur_barberman ORDER BY lemburID ASC");

```

```

$hasil = $query->result_array();
return $hasil;}

public function GetLemburByMonth(){
$query = $this->db->query("SELECT l.*, k.namaDepan FROM lembur_barberman l,
karyawan k WHERE l.karyawanID = k.karyawanID AND MONTH(l.tglLembur) =
MONTH(date(now())) ORDER BY karyawanID");
$hasil = $query->result_array();
return $hasil;}

public function GetByID($lemburID){
$query = $this->db->query("SELECT * FROM lembur_barberman WHERE lemburID =
$lemburID");
$hasil = $query->row_array();
return $hasil;}
//ambil data karyawan yang lembur
public function GetBarbermanLembur($karyawanID, $tglReservasi){
$query = $this->db->query("SELECT * FROM lembur_barberman WHERE karyawanID =
$karyawanID AND tglLembur = '$tglReservasi'");
return $query;}

//batas crud data
public function Simpan($data){
return $this->db->insert($this->table, $data);}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}

public function Hapus($where){
return $this->db->delete($this->table, $where);}?>

```

## Model Libur

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_libur extends CI_Model{
protected $table = 'libur_barberman';
function __construct(){
parent::__construct();}

public function GetAll(){
$query = $this->db->query("SELECT * FROM libur_barberman ORDER BY liburID ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetLiburByMonth(){
$query = $this->db->query("SELECT l.*, k.namaDepan FROM libur_barberman l,
karyawan k WHERE l.karyawanID = k.karyawanID AND MONTH(l.tglLibur) =
MONTH(date(now())) ORDER BY karyawanID");
$hasil = $query->result_array();
return $hasil;}

public function GetByID($liburID){
$query = $this->db->query("SELECT * FROM libur_barberman WHERE liburID =
$liburID");
$hasil = $query->row_array();
return $hasil;}

//batas crud data
public function Simpan($data){
return $this->db->insert($this->table, $data);}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}

public function Hapus($where){
return $this->db->delete($this->table, $where);}?>

```

## Model Pengeluaran

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_pengeluaran extends CI_Model{
protected $table = 'pengeluaran';

```

```

protected $table_detail = 'detail_pengeluaran';
function __construct(){
parent::__construct();}

public function GetByDate(){
$query = $this->db->query("SELECT * FROM pengeluaran WHERE tglPengeluaran =
date(now()) ORDER BY pengeluaranID ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetByDetailID($pengeluaranID){
$query = $this->db->query("SELECT * FROM pengeluaran WHERE pengeluaranID =
$pengeluaranID");
$hasil = $query->row_array();
return $hasil;}

public function GetTotal(){
$query = $this->db->query("SELECT SUM(dp.harga) FROM pengeluaran p,
detail_pengeluaran dp WHERE p.tglPengeluaran = date(now()) AND dp.pengeluaranID =
p.pengeluaranID");
$hasil = $query->row_array();
return $count = $hasil['SUM(dp.harga)'];}

public function GetByID($pengeluaranID){
$query = $this->db->query("SELECT n.*, service_name FROM `pengeluaran_detail` n,
layanan_pengeluaran k
where n.es_code=k.es_code AND n.pengeluaranID=$pengeluaranID ORDER BY
pengeluaranID DESC");
$hasil = $query->result_array();
return $hasil;}

public function GetDetailByDate(){
$query = $this->db->query("SELECT p.pengeluaranID, p.tglPengeluaran, dp.detailID,
dp.karyawanID, k.namaDepan, lp.namaLayanan, dp.harga, dp.keterangan from
pengeluaran p, detail_pengeluaran dp, layanan_pengeluaran lp, karyawan k where
dp.pengeluaranID = p.pengeluaranID and dp.karyawanID = k.karyawanID and
dp.kodeLayanan = lp.kodeLayanan and p.tglPengeluaran = date(now()) group by
p.pengeluaranID, p.tglPengeluaran, dp.detailID, dp.karyawanID, k.namaDepan,
lp.namaLayanan, dp.harga, dp.keterangan");
$hasil = $query->result_array();
return $hasil;}

//batas crud data
public function Simpan($data){
return $this->db->insert($this->table, $data);}
public function SimpanDetail($data){
return $this->db->insert($this->table_detail, $data);}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}

public function Hapus($where){
return $this->db->delete($this->table, $where);}

public function HapusDetail($where){
return $this->db->delete($this->table_detail, $where);}}?>

```

## Model Penggajian

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_penggajian extends CI_Model{
protected $table = 'penggajian';
function __construct(){
parent::__construct();}

public function GetGajian($kryID, $bulan){
$query = $this->db->query("SELECT tglReservasi, namaTreatment, karyawanID,
namaDepan, sum(jumlah) as jumlah, sharing, status, keterangan, sum(subTotal) as
subTotal FROM report_subtotal_penggajian WHERE karyawanID = '$kryID' AND
month(tglReservasi) = $bulan AND keterangan = 'Marked' GROUP BY namaTreatment,
namaDepan, status");
$hasil = $query->result_array();
return $hasil;}

```

```

public function GetTotalGaji($kryID, $bulan){
$query = $this->db->query("SELECT SUM(subTotal) FROM report_subtotal_penggajian
WHERE karyawanID = '$kryID' AND month(tglReservasi) = $bulan AND keterangan =
'Marked'");
$hasil = $query->row_array();
return $count = $hasil['SUM(subTotal)'];}
public function GetLiburByMonth(){
$query = $this->db->query("SELECT l.*, k.namaDepan FROM libur_barberman l,
karyawan k WHERE l.karyawanID = k.karyawanID AND MONTH(l.tglLibur) =
MONTH(date(now())) ORDER BY karyawanID");
$hasil = $query->result_array();
return $hasil;}

public function GetByID($liburID){
$query = $this->db->query("SELECT * FROM libur_barberman WHERE liburID =
$liburID");
$hasil = $query->row_array();
return $hasil;}

//batas crud data
public function Simpan($data){
return $this->db->insert($this->table, $data);}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}

public function Hapus($where){
return $this->db->delete($this->table, $where);}??>

```

## Model Produk

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_produk extends CI_Model{
protected $table = 'produk';
function __construct(){
parent::__construct();}

function GetAll($num, $offset){
$query = $this->db->get("produk", $num, $offset);
return $query->result_array();}
function GetAllProduk(){
$query = $this->db->query("SELECT * FROM produk ORDER BY RAND() LIMIT 10");
return $query->result_array();}

public function GetByKode($kodeProduk){
$query = $this->db->query("SELECT * FROM produk WHERE kodeProduk =
'$kodeProduk'");
$hasil = $query->row_array();
return $hasil;}

//ambil data tabel produk
public function GetProduk($where= ""){
$data = $this->db->query('select * from produk '.$where);
return $data;}

public function Simpan($table, $data){
return $this->db->insert($table, $data);}

public function Update($table,$data,$where){
return $this->db->update($table,$data,$where);}

public function Hapus($table,$where){
return $this->db->delete($table,$where);}

function Updateproduk($data){
$this->db->where('kodeProduk', $data['kodeProduk']);
$this->db->update('produk', $data);}??>

```

## Model Register

```
<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_register extends CI_Model {
protected $table = 'customer';
function __construct(){
parent::__construct();}

function TambahAkun($data){
$query = $this->db->insert('customer',$data);
return $query;}

function UbahAktivasi($key){
$this->load->database();
$data = array(
'aktif' => 1);
$this->db->where('md5(customerID)', $key);
$this->db->update($this->table, $data);
return true;}}?>
```

## Model Reservasi

```
<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_reservasi extends CI_Model{
protected $table = 'reservasi';
protected $table_treatment = 'r_treatment';
protected $table_produk = 'r_produk';
protected $table_tutup_reservasi = 'tutup_reservasi';
protected $table_sisa_waktu = 'sisa_waktu';
function __construct(){
parent::__construct();}

public function GetReservasiByKode($kodeReservasi){
$query = $this->db->query("SELECT * FROM reservasi WHERE kodeReservasi =
'$kodeReservasi'");
$hasil = $query->row_array();
return $hasil;}

public function CekJadwal($tglReservasi, $waktuReservasi, $karyawanID){
$query = $this->db->query("SELECT r.tglReservasi, r.waktuReservasi, rt.karyawanID
FROM reservasi r, r_treatment rt WHERE rt.kodeReservasi = r.kodeReservasi AND
r.tglReservasi = '$tglReservasi' AND r.waktuReservasi = '$waktuReservasi' AND
rt.karyawanID = $karyawanID");
return $query;}

//cek status reservasi online
public function CekStatus(){
$query = $this->db->query("SELECT * FROM tutup_reservasi WHERE id = 1");
$hasil = $query->row('status');
return $hasil;}

//ambil data tabel r_treatment
public function GetTreatment($where= ""){
$data = $this->db->query('select * from r_treatment '.$where);
return $data;}

public function GetAllBarberman(){
$query = $this->db->query("SELECT * FROM karyawan WHERE level = 'Barberman' ORDER
BY karyawanID ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetReservasiRyan(){
$query = $this->db->query("SELECT DISTINCT r.kodeReservasi, r.namaCustomer,
r.tglReservasi, r.waktuReservasi, r.status, k.namaDepan AS namaBarber FROM
reservasi r, r_treatment rt, karyawan k WHERE rt.karyawanID = k.karyawanID AND
rt.kodeReservasi = r.kodeReservasi AND k.namaDepan = 'Ryan' AND r.status IN
('Menunggu','Sedang Dilayani') AND r.tglReservasi = date(now()) ORDER BY
r.waktuReservasi ASC, r.kodeReservasi ASC");
$hasil = $query->result_array();
return $hasil;}

```

```

public function GetReservasiHeri(){
$query = $this->db->query("SELECT DISTINCT r.kodeReservasi, r.namaCustomer,
r.tglReservasi, r.waktuReservasi, r.status, k.namaDepan AS namaBarber FROM
reservasi r, r_treatment rt, karyawan k WHERE rt.karyawanID = k.karyawanID AND
rt.kodeReservasi = r.kodeReservasi AND k.namaDepan = 'Heri' AND r.status IN
('Menunggu','Sedang Dilayani') AND r.tglReservasi = date(now()) ORDER BY
r.waktuReservasi ASC, r.kodeReservasi ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetReservasiPandu(){
$query = $this->db->query("SELECT DISTINCT r.kodeReservasi, r.namaCustomer,
r.tglReservasi, r.waktuReservasi, r.status, k.namaDepan AS namaBarber FROM
reservasi r, r_treatment rt, karyawan k WHERE rt.karyawanID = k.karyawanID AND
rt.kodeReservasi = r.kodeReservasi AND k.namaDepan = 'Pandu' AND r.status IN
('Menunggu','Sedang Dilayani') AND r.tglReservasi = date(now()) ORDER BY
r.waktuReservasi ASC, r.kodeReservasi ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetReservasiDayat(){
$query = $this->db->query("SELECT DISTINCT r.kodeReservasi, r.namaCustomer,
r.tglReservasi, r.waktuReservasi, r.status, k.namaDepan AS namaBarber FROM
reservasi r, r_treatment rt, karyawan k WHERE rt.karyawanID = k.karyawanID AND
rt.kodeReservasi = r.kodeReservasi AND k.namaDepan = 'Dayat' AND r.status IN
('Menunggu','Sedang Dilayani') AND r.tglReservasi = date(now()) ORDER BY
r.waktuReservasi ASC, r.kodeReservasi ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetByKode($kodeReservasi){
$query = $this->db->query("SELECT * FROM reservasi WHERE kodeReservasi =
'$kodeReservasi'");
$hasil = $query->row_array();
return $hasil;}

public function GetTreatmentByKode($kodeReservasi){
$query = $this->db->query("SELECT rt.rtID, rt.kodeReservasi, k.namaDepan,
t.namaTreatment, rt.harga from r_treatment rt, karyawan k ,treatment t WHERE
rt.kodeReservasi = '$kodeReservasi' AND rt.karyawanID = k.karyawanID AND
rt.kodeTreatment = t.kodeTreatment");
$hasil = $query->result_array();
return $hasil;}

public function GetTotalTreatment($kodeReservasi){
$query = $this->db->query("SELECT SUM(harga) FROM r_treatment WHERE kodeReservasi
= '$kodeReservasi'");
$hasil = $query->row_array();
return $count = $hasil['SUM(harga)'];}

public function GetProdukByKode($kodeReservasi){
$query = $this->db->query("SELECT rp.rpID, rp.kodeReservasi, p.namaProduk, rp.qty,
rp.harga, rp.subTotal FROM r_produk rp, produk p WHERE rp.kodeReservasi =
'$kodeReservasi' AND rp.kodeProduk = p.kodeProduk");
$hasil = $query->result_array();
return $hasil;}

public function GetTotalProduk($kodeReservasi){
$query = $this->db->query("SELECT SUM(subTotal) FROM r_produk WHERE kodeReservasi
= '$kodeReservasi'");
$hasil = $query->row_array();
return $count = $hasil['SUM(subTotal)'];}

//insert table reservasi
public function Simpan($data){
return $this->db->insert($this->table, $data);}

//insert table r_treatment
public function SimpanTreatment($data){
return $this->db->insert($this->table_treatment, $data);}

//update r_treatment
function UpdateTreatment($data){
$this->db->where('rtID', $data['rtID']);

```

```

$this->db->update($this->table_treatment, $data);}

//hapus r_treatment
public function HapusTreatment($table, $where){
return $this->db->delete($table,$where);}

//insert table r_produk
public function SimpanProduk($data){
return $this->db->insert($this->table_produk, $data);}

//hapus r_treatment
public function HapusProduk($table, $where){
return $this->db->delete($table,$where);}

//update reservasi status
public function UpdateStatus($data,$where){
return $this->db->update($this->table, $data, $where);}

//batal reservasi atau hapus
public function Batal($where){
return $this->db->delete($this->table, $where);}

//buka reservasi
public function BukaReservasi($where){
return $this->db->update($this->table_tutup_reservasi, $where);}

//tutup reservasi
public function TutupReservasi($where){
return $this->db->update($this->table_tutup_reservasi, $where);}

//insert table sisa waktu
public function SimpanSisaWaktu($fields){
return $this->db->insert($this->table_sisa_waktu, $fields);}

//Update sisa waktu
public function UpdateSisaWaktu($fields){
$this->db->where('id', $fields['id']);
return $this->db->update($this->table_sisa_waktu, $fields);}

//Update waktu reservasi
public function UpdateWaktuReservasi($atribut){
$this->db->where('kodeReservasi', $atribut['kodeReservasi']);
return $this->db->update($this->table, $atribut);}?>

```

## Model Treatment

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');
class M_treatment extends CI_Model{
protected $table = 'treatment';
function __construct(){
parent::__construct();}

public function GetAll(){
$query = $this->db->query("SELECT * FROM treatment ORDER BY kodeTreatment ASC");
$hasil = $query->result_array();
return $hasil;}

public function GetByKode($kodeTreatment){
$query = $this->db->query("SELECT * FROM treatment WHERE kodeTreatment = '$kodeTreatment'");
$hasil = $query->row_array();
return $hasil;}

public function save($data){
$fields = array(
'kodeTreatment' => $data['kodeTreatment'],
'namaTreatment' => $data['namaTreatment'],
'harga' => $data['harga'],
'durasi' => $data['durasi'],
'keterangan' => $data['keterangan'],
'gambar' => $data['gambar'],);

```

```

// check if data existed
$condition = array(
'kodeTreatment' => $data['kodeTreatment'],);
$query = $this->db->get_where($this->table, $condition);
$result = $query->result_array();
if(!empty($result)){
//proses update table
$this->db->where($condition);
$this->db->update($this->table, $fields);}else{
// proses insert table
return $this->db->insert($this->table, $fields);} }

public function Hapus($table,$where){
return $this->db->delete($table,$where);}

public function Simpan($data){
return $this->db->insert($this->table, $data);}

public function Update($data,$where){
return $this->db->update($this->table, $data, $where);}?>

```

## View Head CSS

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta name="description" content="">
<meta name="author" content="EGrappler.com, Sarfraz Shoukat, Muhammad Shahbaz Saleem">
<link rel="shortcut icon" href="<?php echo base_url('images/ico/favicon.png')?>">
<title><?php echo $title ?></title>
<!-- Bootstrap core CSS -->
<link href="<?php echo base_url().'css/bootstrap.css'?>" rel="stylesheet">
<link href="<?php echo base_url().'css/custom.css'?>" rel="stylesheet">
<!-- Custom styles for this template -->
<link href="http://netdna.bootstrapcdn.com/font-awesome/4.6.2/css/font-awesome.css" rel="stylesheet">
<link href="<?php echo base_url().'css/style.css'?>" rel="stylesheet">
<link href="<?php echo base_url().'css/jquery.bxslider.css'?>" rel="stylesheet" type="text/css" />
<link href="<?php echo base_url().'css/tipsy.css'?>" rel="stylesheet">
<script type="text/javascript" src="<?php echo base_url().'js/jquery-1.11.1.min.js'?>"></script>
<script type="text/javascript" src="<?php echo base_url().'js/modernizr.custom.js'?>"></script>
<link href="<?php echo base_url().'plugins/datatables/dataTables.bootstrap.css'?>" rel="stylesheet">

```

## View Head Menu

```

<header class="header">
<div class="container">
<div class="row">
<div class="col-md-3">
<div id="dl-menu-logo" class="dl-menuwrapper-logo">
<div class="logo">
<a href="<?php echo base_url().'office/main'?>">
<!--
>
MTB</a></div></div></div>
<div class="col-md-12">
<div class="menu">
<div id="dl-menu" class="dl-menuwrapper">
<button class="dl-trigger">Open Menu</button>
<ul class="dl-menu">
<li>
<?php if(!empty($user)): ?>
<?php if($user['level'] == "Owner"): ?>
<a href="#"> Setup</a>
<ul class="dl-submenu">
<li><a href="<?php echo site_url().'office/karyawan'?>">Karyawan</a></li>
<li><a href="<?php echo site_url().'office/libur'?>">Libur Barberman</a></li>
<li><a href="<?php echo site_url().'office/lembur'?>">Lembur Barberman</a></li>

```



```

<li><a href="<?php echo site_url().'office/gaji'?">Gaji Barberman</a></li>
<li><a href="<?php echo site_url().'office/penggajian'?">Gajian Barberman</a></li>
<li><a href="<?php echo site_url().'office/treatment'?">Treatment</a></li>
<li><a href="<?php echo site_url().'office/produk'?">Produk</a></li>
<li><a href="<?php echo site_url().'office/gayarambut'?">Gaya Rambut</a></li>
<li><a href="<?php echo site_url().'office/pengeluaran'?">Pengeluaran</a></li>
</ul>
<li><a href="<?php echo site_url().'office/laporan'?">Laporan</a></li>
<li><a href="<?php echo site_url().'office/reservasi'?">Reservasi</a></li>
<li><a href="<?php echo site_url().'office/blog'?">Blog</a></li>
<li><a href="<?php echo site_url().'office/gallery'?">Gallery</a></li>
<li><a href="<?php echo site_url().'office/about'?">Tentang Kami</a></li>
<?php elseif($user['level'] == "Manager"): ?>
<a href="#">Setup</a>
<ul class="dl-submenu">
<li><a href="<?php echo site_url().'office/treatment'?">Treatment</a></li>
<li><a href="<?php echo site_url().'office/produk'?">Produk</a></li>
<li><a href="<?php echo site_url().'office/gayarambut'?">Gaya Rambut</a></li>
<li><a href="<?php echo site_url().'office/pengeluaran'?">Pengeluaran</a></li>
</ul>
<li><a href="<?php echo site_url().'office/laporan'?">Laporan</a></li>
<li><a href="<?php echo site_url().'office/reservasi'?">Reservasi</a></li>
<li><a href="<?php echo site_url().'office/blog'?">Blog</a></li>
<li><a href="<?php echo site_url().'office/gallery'?">Gallery</a></li>
<li><a href="<?php echo site_url().'office/about'?">Tentang Kami</a></li>
<?php elseif($user['level'] == "Cashier"): ?>
<a href="#">Setup</a>
<ul class="dl-submenu">
<li><a href="<?php echo site_url().'office/treatment'?">Treatment</a></li>
<li><a href="<?php echo site_url().'office/produk'?">Produk</a></li>
<li><a href="<?php echo site_url().'office/pengeluaran'?">Pengeluaran</a></li>
</ul>
<li><a href="<?php echo site_url().'office/reservasi'?">Reservasi</a></li>
<li><a href="<?php echo site_url().'office/blog'?">Blog</a></li>
<li><a href="<?php echo site_url().'office/gallery'?">Gallery</a></li>
<li><a href="<?php echo site_url().'office/about'?">Tentang Kami</a></li>
<?php endif ?>
<?php endif ?>
</li>
<?php if($this->session->userdata('logged_in')): ?>
<li><a href="<?php echo site_url().'office/akun/profile'?"><?php echo $user['namaDepan']?"></a></li>
<li><a href="<?php echo site_url().'mark/logout'?">Logout</a></li>
<?php else: ?>
<a href="<?php echo site_url().'authentication'?">Login</a>
<?php endif ?>
</ul></div> </div><!-- /.menu --> </div></div></div>
</header>

```

## View Footer

```

<footer>
<div class="footer-top">
<div class="container">
<div class="row">
<div class="col-md-8">
<h3>MARK THE BARBER!</h3>
<p></p><p class="copyright">&copy; <?php $this->load->helper('date'); $datestring = "%Y"; echo mdate($datestring)?? All Rights Reserved. Templated by <a href="https://www.egrappler.com" target="_blank">EGrappler</a></p>
<p class="copyright">Developed by <a href="https://www.instagram.com/orkartanda/" target="_blank">Hartadi Kartanda</a></p>
</div><div class="col-md-4">
<h3>KONTAK KAMI</h3>
<p>Jl. Kaliurang, gg. Pandega Karya No.290, Depok, Kecamatan Sleman.</p>
<p>Daerah Istimewa Yogyakarta, Indonesia.</p>
<ul class="get-social"><li>
<a href="https://www.facebook.com/markthebarber.yk" target="_blank">
<i class="fa fa-facebook"></i>
</a></li>
<li>
<a href="https://twitter.com/markthebarberyk" target="_blank">

```

```

<i class="fa fa-twitter"></i></a></li>
<li><a href="https://www.instagram.com/markthebarber_yk/" target="_blank">
<i class="fa fa-instagram"></i></a></li></ul></div></div></div></div>
</footer>

```

## View Javascript

```

<script type="text/javascript" src="<?php echo
base_url(). 'js/bootstrap.js'?>"></script>
<script type="text/javascript" src="<?php echo
base_url(). 'js/jquery.dlmenu.js'?>"></script>
<script type="text/javascript" src="<?php echo
base_url(). 'js/jquery.flexisel.js'?>"></script>
<script type="text/javascript" src="<?php echo
base_url(). 'js/jquery.prettyPhoto.js'?>"></script>
<script type="text/javascript" src="<?php echo
base_url(). 'js/jquery.easing.1.3.js'?>"></script>
<script type="text/javascript" src="<?php echo base_url(). 'js/jquery.ajax-contact-
form.js'?>"></script>
<script type="text/javascript" src="<?php echo
base_url(). 'js/script.js'?>"></script>
<script type="text/javascript" src="<?php echo
base_url(). 'js/jquery.tipsy.js'?>"></script>
<script type="text/javascript" src="<?php echo
base_url(). 'js/jquery.bxslider.js'?>"></script>
<script src="<?php echo base_url(); ?>plugins/jquery/jquery-
2.1.3.min.js"></script>
<script src="<?php echo base_url(); ?>dist/js/jquery-ui.min.js"
type="text/javascript"></script>
<script>$.widget.bridge('uibutton', $.ui.button);</script>
<script src="<?php echo base_url(); ?>plugins/datepicker/bootstrap-datepicker.js"
type="text/javascript"></script>
<script src="<?php echo base_url(); ?>plugins/iCheck/ichack.min.js"
type="text/javascript"></script>
<script src="<?php echo base_url(); ?>plugins/slimScroll/jquery.slimscroll.min.js"
type="text/javascript"></script>
<script src='<?php echo base_url();
?>plugins/fastclick/fastclick.min.js'></script>
<script src="<?php echo base_url(); ?>dist/js/app.min.js"
type="text/javascript"></script>

```

**LAMPIRAN B: DOKUMEN-DOKUMEN TERKAIT**

1 PHS (5,90)	D 25.000	Get - 25.000
2 PHS (8,73)	D 25.000	laundry - 20.000
3 PHS (11,46)	D 25.000	gaji Barbers 100.000
4 PHS (11,75)	D 25.000	Pinus 100.000
5 PHS (11,94)	D 25.000	D = PHS - 8
6 PHS (12,39)	D 25.000	DR = 3
7 PHS (13,40)	D 25.000	A = PHS - 0
8 PHS (14,38)	D 25.000	HS = PHS - 4
9 PHS (15,34)	H 25.000	DR = 2
10 PHS (16,20)	F 25.000	KIOS 1
11 PHS (17,13)	R 25.000	SHOUBH
12 PHS (18,00)	F 25.000	F = PHS - 1
13 PHS (18,98)	H 25.000	DR = 2
14 PHS (19,95)	D 25.000	PHS FREE - 1
15 PHS (20,90)	H 25.000	PHS - 35 x 25.000
16 PHS (21,83)	R 25.000	= 1.225.000
17 PHS (22,74)	R 25.000	DR = 2 x 25.000
18 PHS (23,62)	H 25.000	= 50.000
19 PHS (24,54)	H 25.000	KIOS - 1 x 25.000
20 PHS (25,40)	D 25.000	= 25.000
21 PHS (26,25)	D 25.000	SHOUBH - 1 x 25.000
22 PHS (27,09)	H 25.000	= 25.000
23 PHS (27,92)	H 25.000	SHOUBH CHOK 200.000
24 PHS (28,74)	D 25.000	
25 PHS (29,55)	D 45.000	

Gambar Data Transaksi Mark The Barber (1)

26 PHS (30,35)	R 25.000	
27 PHS (31,14)	R 25.000	
28 PHS (31,92)	R 25.000	
29 PHS (32,69)	H 25.000	
30 PHS (33,45)	P 200.000	
31 PHS (34,20)	P 25.000	
32 PHS (34,95)	H 25.000	
33 PHS (35,69)	H 25.000	
34 PHS (36,42)	R 25.000	
35 PHS (37,14)	H 25.000	
36 PHS (37,85)	H 25.000	
37 PHS (38,55)	R 25.000	
38 PHS (39,24)	P 25.000	
39 PHS (39,92)	R 25.000	
40 PHS (40,59)	H 25.000	
41 PHS (41,25)	H 25.000	
42 PHS (41,90)	R 25.000	
43 PHS (42,54)	P 25.000	
44 PHS (43,17)	H 25.000	
45 PHS (43,79)	P 25.000	
46 PHS (44,40)	H 25.000	
47 PHS (45,00)	R 25.000	
48 PHS (45,59)	H 25.000	
49 PHS (46,17)	P 25.000	
50 PHS (46,74)	H 25.000	
51 PHS (47,30)	R 25.000	
52 PHS (47,85)	H 25.000	
53 PHS (48,39)	P 25.000	
54 PHS (48,92)	H 25.000	
55 PHS (49,44)	R 25.000	
56 PHS (49,95)	H 25.000	
57 PHS (50,45)	R 25.000	
58 PHS (50,94)	P 25.000	
59 PHS (51,42)	H 25.000	
60 PHS (51,89)	R 25.000	
61 PHS (52,35)	H 25.000	
62 PHS (52,80)	P 25.000	
63 PHS (53,24)	H 25.000	
64 PHS (53,67)	R 25.000	
65 PHS (54,09)	H 25.000	
66 PHS (54,50)	P 25.000	
67 PHS (54,90)	H 25.000	
68 PHS (55,29)	R 25.000	
69 PHS (55,67)	H 25.000	
70 PHS (56,04)	P 25.000	
71 PHS (56,40)	H 25.000	
72 PHS (56,75)	R 25.000	
73 PHS (57,09)	H 25.000	
74 PHS (57,42)	P 25.000	
75 PHS (57,74)	H 25.000	
76 PHS (58,05)	R 25.000	
77 PHS (58,35)	H 25.000	
78 PHS (58,64)	P 25.000	
79 PHS (58,92)	H 25.000	
80 PHS (59,19)	R 25.000	
81 PHS (59,45)	H 25.000	
82 PHS (59,70)	P 25.000	
83 PHS (59,94)	H 25.000	
84 PHS (60,17)	R 25.000	
85 PHS (60,39)	H 25.000	
86 PHS (60,60)	P 25.000	
87 PHS (60,80)	H 25.000	
88 PHS (61,00)	R 25.000	
89 PHS (61,19)	H 25.000	
90 PHS (61,37)	P 25.000	
91 PHS (61,54)	H 25.000	
92 PHS (61,70)	R 25.000	
93 PHS (61,85)	H 25.000	
94 PHS (62,00)	P 25.000	
95 PHS (62,14)	H 25.000	
96 PHS (62,27)	R 25.000	
97 PHS (62,40)	H 25.000	
98 PHS (62,52)	P 25.000	
99 PHS (62,63)	H 25.000	
100 PHS (62,74)	R 25.000	

Gambar Data Transaksi Mark The Barber (2)

Nama	Tanggal	Waktu	Barberman	Treatment	Harga	Sharing	Status
Customer	2016-05-29	14:58:00	Ryan	Premium Haircut Service for Adults	35000	10500	Regular
Customer	2016-05-29	15:40:00	Ryan	Premium Haircut Service for Adults	35000	10500	Regular
Customer	2016-05-29	15:58:00	Ryan	Premium Haircut Service for Adults	35000	10500	Regular
Customer	2016-05-29	16:58:00	Ryan	Premium Haircut Service for Adults	35000	10500	Regular
Customer	2016-05-29	13:34:00	Heri	Premium Haircut Service for Adults	35000	9000	Regular
Customer	2016-05-29	15:09:00	Heri	Premium Haircut Service for Adults	35000	9000	Regular
Customer	2016-05-29	15:57:00	Heri	Premium Haircut Service for Adults	35000	9000	Regular
Customer	2016-05-29	16:52:00	Heri	Premium Haircut Service for Adults	35000	9000	Regular
Customer	2016-05-29	14:00:00	Pandu	Premium Haircut Service for Adults	35000	8000	Regular
Customer	2016-05-29	15:06:00	Pandu	Premium Haircut Service for Adults	35000	8000	Regular
Customer	2016-05-29	16:00:00	Pandu	Premium Haircut Service for Adults	35000	8000	Regular
Customer	2016-05-29	17:00:00	Pandu	Premium Haircut Service for Adults	35000	8000	Regular
Customer	2016-05-29	09:40:00	Dayat	Premium Haircut Service for Adults	35000	7000	Regular
Customer	2016-05-29	10:17:00	Dayat	Premium Haircut Service for Adults	35000	7000	Regular
Customer	2016-05-29	10:45:00	Dayat	Premium Haircut Service for Adults	35000	7000	Regular
Customer	2016-05-29	11:12:00	Dayat	Premium Haircut Service for Adults	35000	7000	Regular
Customer	2016-05-29	12:50:00	Dayat	Premium Haircut Service for Adults	35000	7000	Regular
Customer	2016-05-29	13:28:00	Dayat	Premium Haircut Service for Adults	35000	7000	Regular
Customer	2016-05-29	11:40:00	Dayat	Dry Cut	25000	5000	Regular
Customer	2016-05-29	12:00:00	Dayat	Dry Cut	25000	5000	Regular
Customer	2016-05-29	16:00:00	Pandu	Traditional Shaving	15000	3500	Regular

**Gambar Data Transaksi Sistem Mark The Barber**

## DAFTAR PUSTAKA

- Aryani, D., Setiadi, A., Adnandi, M. A., Nurjaman, I., & Widiarso, R. T. (2014, Desember 6). Perancangan Sistem Informasi Pelayanan Klinik Menggunakan Model Antrian First In First Out. *Seminar Nasional Informatika Medis (SNIMed)*, *V*, 22.
- B, J. E., & Angwarmasse, W. (2012). Model Antrian FIFO (First In First Out) Pada Pelayanan Mahasiswa Fakultas Teknik Universitas Janabadra Berbasis Multimedia. *Jurnal Teknik*, *2*.
- Christanto, W., Prasida, A. S., & Fibriani, C. (2012, Januari). Perancangan dan Implementasi Sistem Reservasi Foodcourt Berbasis Web dengan Memanfaatkan Koneksi Wifi. *Jurnal Buana Informatika*, *3*, 39-50.
- Fahrudin, A., Purnama, B. E., & Riasti, B. K. (2011). Pembangunan Sistem Informasi Layanan Haji Berbasis Web. *Journal Speed – Sentra Penelitian Engineering dan Edukasi*, *3*, 36.
- Wibowo. 2014. “Rancang Bangun Sistem Informasi Reservasi Sewa Kamar Hotel Berbasis Web (Studi Kasus Hotel Green Mandarin Pekalongan)”. *Skripsi*. Fakultas Teknologi Informasi, Program Studi Teknik Informatika, Universitas STIKUBANK (UNISBANK) Semarang.
- Yuliawan, Y., Sunarto, M. D., & Soebijono, T. (2013). Pengembangan Sistem Informasi Pendataan Jemaat Gereja Masehi Advent Hari Ketujuh Konferens Jawa Kawasan Timur Berbasis Web. *JSIKA*, *2*, 85. Retrieved from <http://jurnal.stikom.edu/index.php/jsika>