

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

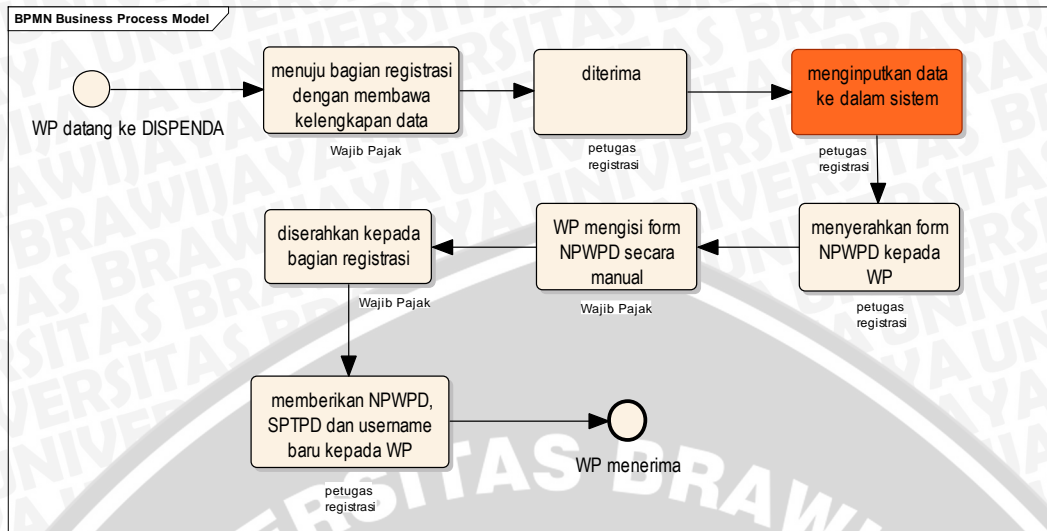
2.1 Kajian Pustaka

Kajian pustaka yang digunakan pada penelitian ini adalah “Analisis dan Perancangan Proses Bisnis Pajak Perhotelan Online Menggunakan ‘Zachman Framework’ (Studi Kasus di DISPENDA Kota Malang)”. Mengacu pada penelitian sebelumnya, penelitian ini melanjutkan implementasi dari hasil proses bisnis yang di rancang pada penelitian sebelumnya. Berikut hasil proses bisnis yang dihasilkan dari penelitian sebelumnya.

2.1.1. Pendaftaran Hotel Baru

Proses ini dilakukan ketika terdapat wajib pajak baru ingin mendaftarkan hotelnya kepada DISPENDA Kota Malang. Tidak ada banyak perbedaan antara proses sebelum dengan sesudah menggunakan sistem, perbedaan hanya terdapat pada proses dimana petugas registrasi menginputkan seluruh data ke dalam sistem. Terdapat dua aktor yang terlibat dalam proses ini yaitu, WP dan petugas registrasi dari DISPENDA. Berikut penjelasan prosesnya :

- WP datang ke DISPENDA.
- WP menuju petugas registrasi dengan membawa kelengkapan data.
- Diterima petugas registrasi.
- Petugas registrasi menginputkan data ke dalam sistem.
- Menyerahkan *form* NPWPD kepada WP.
- WP mengisi *form* NPWPD secara manual.
- Diserahkan kepada petugas registrasi.
- Petugas registrasi akan memberikan NPWPD, SPTPD dan *username* baru kepada WP.
- WP menerima.

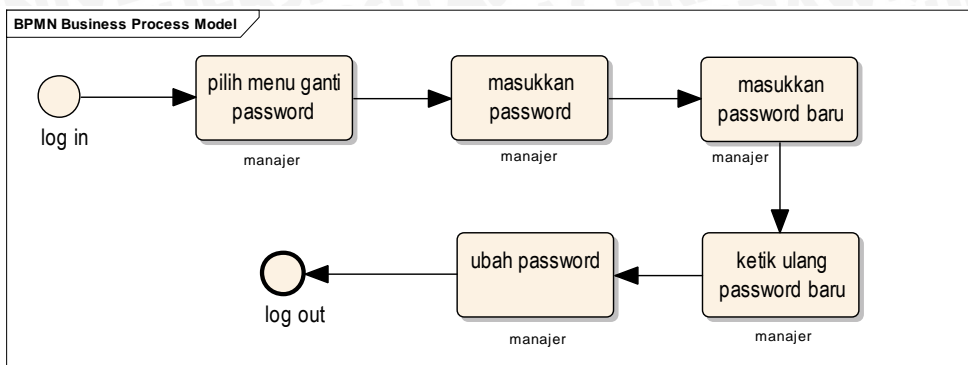


Gambar 2.1 Alur Proses Baru Tambah Hotel Baru

2.1.2. Inisialisasi Awal

Setelah WP mendaftarkan usahanya ke DISPENDA maka WP akan mendapatkan *Username* dan *password* untuk mengakses sistem. *Username* dan *password* yang diberikan kepada manajer hotel masih dalam bentuk standar yaitu, *username* dengan “nama hotel” dan *password* dengan “nama hotel. Berikut adalah urutan proses inisialisasi awal:

- *User* (manajer) *log in* menggunakan *username* dan *password* yang telah diberikan sebelumnya.
- Pilih menu ganti *password*.
- Masukkan *password* awal.
- Masukkan *password* baru.
- Ketik ulang *password* baru.
- Ubah *password*.
- *Logout*.

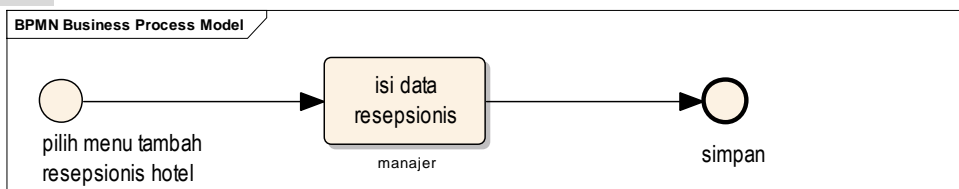


Gambar 2.2 Alur Proses Baru Inisialisasi Awal

2.1.3. Tambah Akun

Proses ini ditujukan agar resepsionis memiliki hak akses untuk menggunakan sistem selain manajer hotel itu sendiri. Proses tambah akun hanya dapat dilakukan oleh satu aktor yang berwenang yaitu manajer hotel. Berikut alurnya:

- Pilih menu tambah resepsionis hotel.
- Isi data resepsionis (no pegawai, nama dan data diri lainnya).
- Simpan.



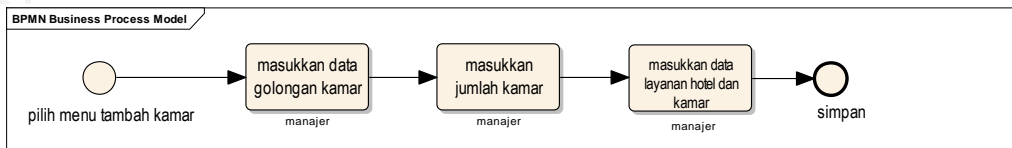
Gambar 2.3 Alur Proses Baru Tambah Akun

2.1.4. Input Data Hotel

Proses ini dilakukan untuk menginputkan data kamar dan layanan hotel seperti jumlah kamar yang disewakan, golongan kamar dan berbagai macam fasilitas (spa, laundry, mini bar, dll). Manajer dan resepsionis dapat mengakses sistem ini. Berikut merupakan alur proses dari *input* data hotel :

- Pilih menu tambah kamar.
- Masukkan data golongan kamar (*superior, deluxe, suite, dll*) yang terdapat di dalam hotel.
- Masukkan jumlah kamar yang terdapat di dalam hotel.

- Masukkan data layanan hotel dan kamar (*laundry, mini bar, restaurant, spa, dll*) yang terdapat di dalam hotel.
- Simpan.

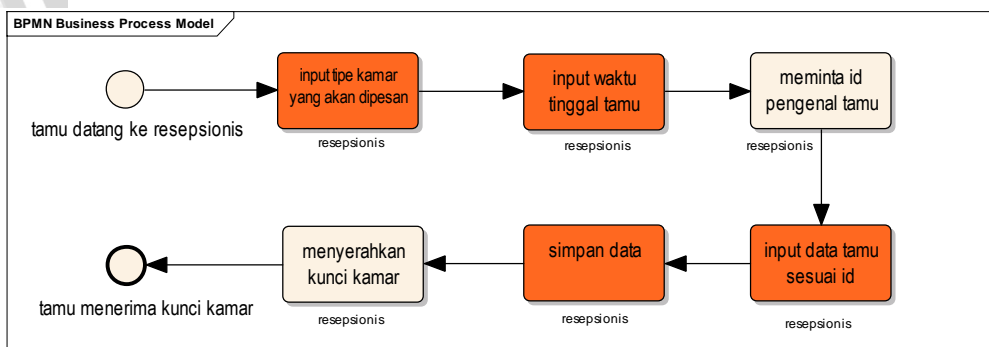


Gambar 2.4 Alur Proses Baru Input Data Hotel

2.1.5. Check In

Proses *check in* yang dilakukan setelah ada sistem tidak jauh berbeda dengan proses *check in* sebelumnya. Perbedaan hanya terdapat pada pencatatan dan penyimpanan data yang langsung diinputkan ke dalam sistem. Berikut adalah proses *check-in* nya:

- Tamu datang ke meja resepsionis
- Resepsionis menginputkan tipe kamar yang akan dipesan.
- Resepsionis menginputkan berapa lama tamu akan tinggal.
- Resepsionis meminta id pengenalan tamu.
- Input data tamu sesuai id pengenalan
- Simpan data yang telah diinputkan.
- Resepsionis menyerahkan kunci kamar kepada tamu.
- Menerima kunci kamar



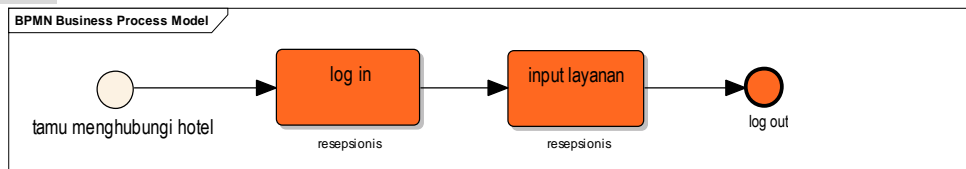
Gambar 2.5 Alur Proses Baru *Check-In*



2.1.6. Tambah Layanan

Proses tambah layanan dilakukan ketika terdapat ini menjelaskan ketika terdapat tamu yang ingin menggunakan layanan kamar setelah melakukan proses *check in* di meja resepsionis dan dapat dilakukan oleh dua aktor yaitu resepsionis dan manajer hotel. Perbedaan yang terjadi dengan proses yang sebelumnya adalah seluruh permintaan akan diinputkan langsung ke dalam sistem. Berikut alur tambah layanan:

- Tamu menghubungi hotel
- *User* melakukan *login* pada sistem
- Input pilihan layanan.
- *Logout*

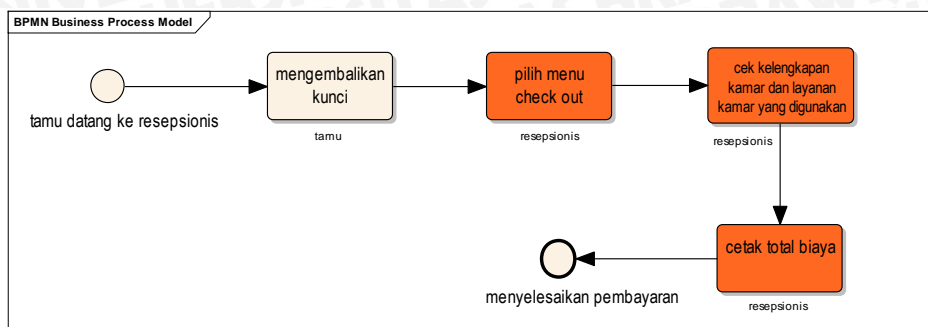


Gambar 2.6 Alur Proses Baru Tambah Layanan Kamar

2.1.7. Check Out

Proses *check out* dilakukan ketika tamu telah selesai menggunakan seluruh fasilitas hotel. Perbedaan dengan sistem sebelumnya yaitu, transaksi dan pengecekan total biaya berlangsung pada sistem. Proses ini dapat dilakukan oleh dua aktor yaitu resepsionis dan manajer hotel. Berikut alur prosesnya:

- Tamu datang ke resepsionis.
- Tamu mengembalikan kunci.
- Pilih menu *check out*.
- Cek kelengkapan dan layanan hotel/kamar yang digunakan
- Cek total biaya
- Tamu menyelesaikan pembayaran

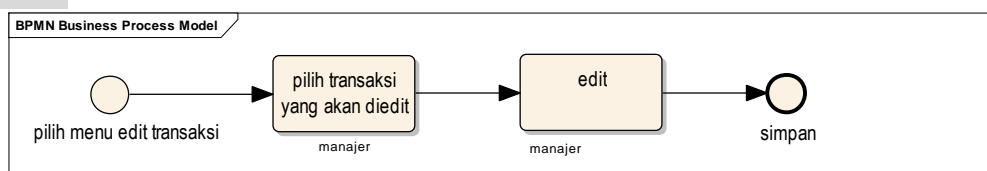


Gambar 2.7 Alur Proses Baru *Check Out*

2.1.8. Edit Transaksi

Proses ini dilakukan ketika terjadi kesalahan atau *human error* yang disebabkan oleh resepsionis atau manajer hotel dalam melaukan transaksi. Fitur edit transaksi ini hanya dapat dilakukan max 1x24 jam setelah terjadinya transaksi agar mengurangi kecurangan yang dapat terjadi. Berikut merupakan alur *edit* transaksi setelah menggunakan sistem:

- Pilih menu *edit* transaksi.
- Pilih transaksi yang akan diedit.
- Edit transaksi.
- Simpan

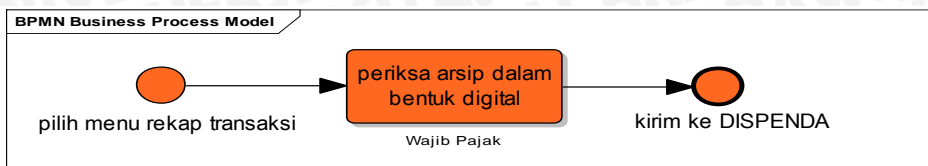


Gambar 2.8 Alur Proses Baru Edit Transaksi

2.1.9. Pengiriman SPTPD

Proses pembayaran ini dimulai dari pengiriman SPTPD kepada DISPENDA dan hanya dapat dilakukan satu aktor yaitu Manajer Hotel saja. Berikut penjelasannya:

- Pilih menu rekap transaksi.
- Periksa arsip dalam bentuk *digital*.
- Kirim ke DISPENDA.

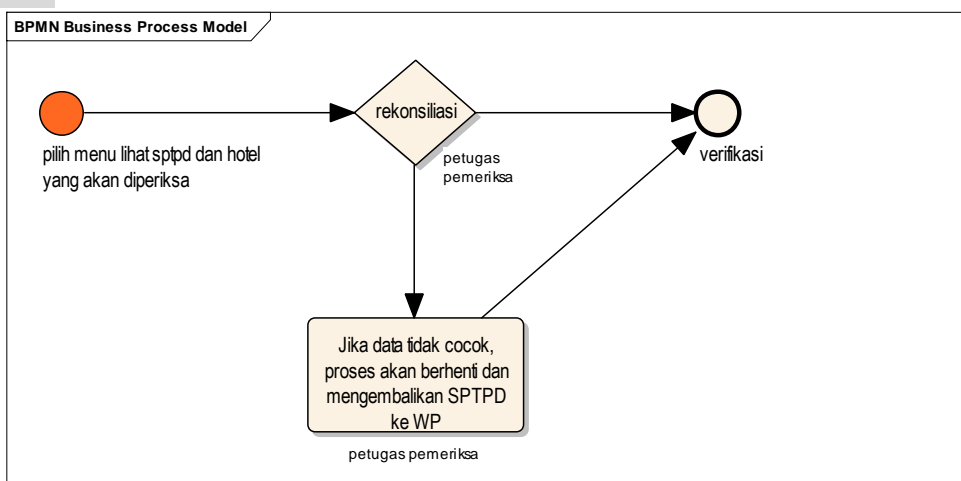


Gambar 2.9 Alur Proses Baru Pengiriman SPTPD

2.1.10. Pemeriksaan SPTPD

Pemeriksaan SPTPD dilakukan untuk rekonsiliasi dan dilakukan satu aktor yaitu petugas pemeriksa berkas. Pada proses ini terdapat proses rekonsiliasi yang berarti pencocokan data transaksi yang dimiliki oleh DISPENDA dan wajib pajak. Setelah proses rekonsiliasi terdapat verifikasi sebagai pernyataan kebenaran laporan yang disetujui oleh kepala seksi, kepala bidang dan kepala dinas. Berikut alurnya:

- Pilih menu lihat SPTPD dan hotel yang akan diperiksa.
- Petugas melakukan rekonsiliasi.
- Jika data cocok, proses berlanjut untuk diverifikasi.
- Jika data tidak cocok, proses akan diberhentikan dan mengembalikan SPTPD ke WP.
- Verifikasi SPTPD.

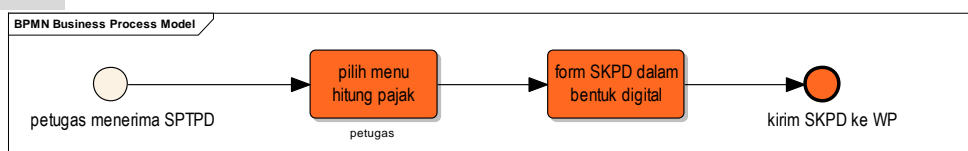


Gambar 2.10 Alur Proses Baru Pemeriksaan SPTPD

2.1.11. Penetapan SKPD

Proses selanjutnya adalah penetapan SKPD. Proses ini hanya dapat diakses oleh satu aktor yaitu petugas bidang pajak daerah lainnya untuk menetapkan SKPD. Perhitungan pajak akan dilakukan secara otomatis oleh sistem sesuai peraturan daerah yaitu, 10% per bulan dan jika terjadi keterlambatan makan setiap bulannya akan ditambahkan 2% per bulan. Berikut merupakan alur dari penetapan SKPD:

- Petugas menerima SPTPD yang telah diverifikasi.
- Pilih menu hitung pajak (sistem akan menghitung nilai pajak sesuai peraturan dan ketentuan).
- Menghasilkan *form* SKPD dalam bentuk *digital*.
- Kirim SKPD ke WP.

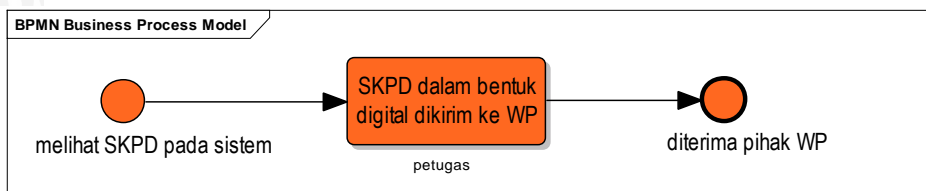


Gambar 2.11 Alur Proses Baru Penetapan SKPD

2.1.12. Kirim SKPD

Proses ini dilakukan setelah SKPD mendapat verifikasi dari kepala seksi, kepala bidang dan kepala dinas. Pengiriman SKPD dilakukan oleh Petugas Bidang Pajak Daerah Lainnya dan berikut alur pengirimannya:

- Melihat SKPD di aplikasi
- *Form* SKPD dalam bentuk digital dikirim ke WP
- Diterima oleh pihak WP

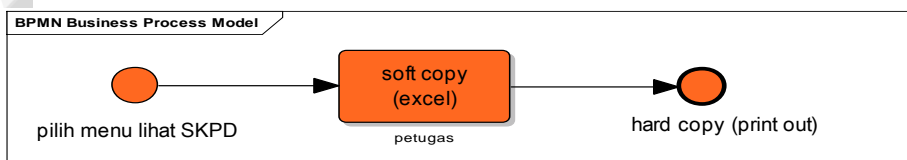


Gambar 2.12 Alur Proses Baru Kirim SKPD

2.1.13. Penerimaan SKPD

Ketika WP sudah menerima SKPD, WP dapat mengakses sistem untuk mengeceknya. Jika WP telah menerima dan memeriksa SKPD maka akan muncul notifikasi pada sistem yang terdapat pada DISPENDA. Terdapat dua pilihan untuk melihat SKPD yang telah diterima yaitu secara *hard copy (print out)* atau *soft copy (export dalam excel)*. Berikut merupakan alur penerimaannya:

- Pilih menu lihat SKPD.
- *Soft copy* dalam bentuk *excel*.
- *Hard Copy* dalam bentuk *print out*.

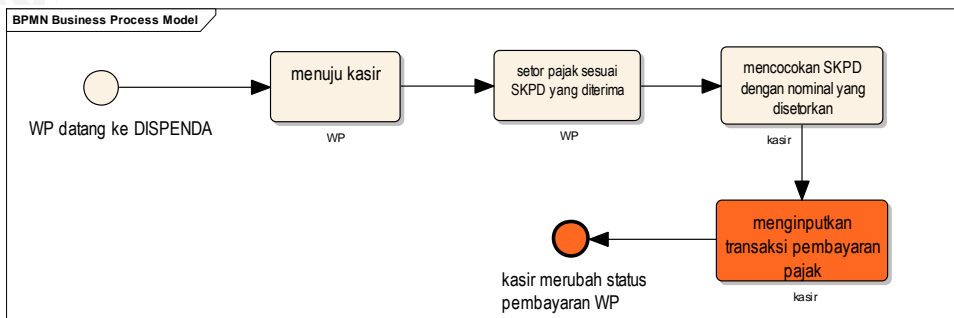


Gambar 2.13 Alur Proses Baru Penerimaan SKPD

2.1.14. Pembayaran

Pembayaran pajak langsung dilakukan dengan membayar total tagihan ke kasir DISPENDA Kota Malang. Pembayaran melalui kasir DISPENDA:

- WP datang ke DISPENDA.
- Menuju kasir.
- WP setor pajak sesuai SKPD.
- Kasir mencocokkan SKPD dengan nominal uang yang disetorkan.
- Menginputkan transaksi pembayaran ke dalam sistem (nominal yang dibayarkan, tanggal bayar, dll).
- Kasir merubah status pembayaran WP pada sistem.

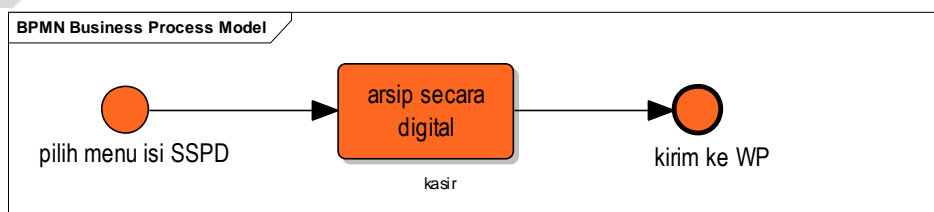


Gambar 2.14 Alur Proses Baru Pembayaran Melalui Kasir

2.1.15. SSPD

Proses ini berlangsung ketika WP telah melaksanakan kewajibannya membayar pajak dan mendapatkan SSPD sebagai tanda bukti pembayaran yang sah. Pengisian SSPD ini dilakukan oleh kasir atau bendahara DISPENDA. Berikut alur prosesnya:

- Pilih menu isi SSPD.
- Arsip secara digital.
- Kirim ke WP.

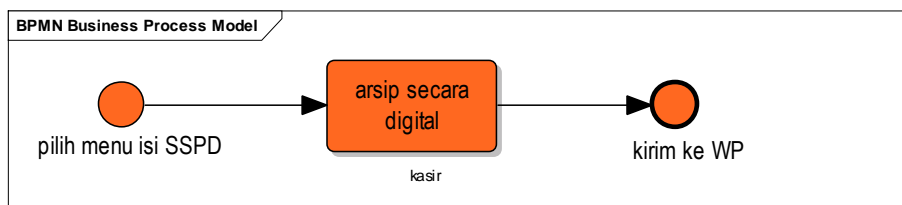


Gambar 2.15 Alur Proses Baru Pengisian SSPD

2.1.16. Pengisian Kartu Data

Proses ini berlangsung ketika WP telah melaksanakan kewajibannya membayar pajak dan mendapatkan SSPD sebagai tanda bukti pembayaran yang sah. Pengisian SSPD ini dilakukan oleh kasir atau bendahara DISPENDA. Berikut alur prosesnya:

- Pilih menu isi SSPD.
- Arsip secara digital.
- Kirim ke WP.



Gambar 2.16 Alur Proses Baru Pengisian SSPD

2.2 Dasar Teori

Pada tahap ini akan dijelaskan dasar teori yang digunakan dalam penelitian ini.

2.2.1. Pendapatan Daerah

Menurut Undang-Undang No.32 tahun 2004 pasal 1 tentang Pemerintah Daerah. Pendapatan daerah adalah semua hak daerah yang diakui sebagai penambah nilai kekayaan bersih dalam periode tahun anggaran yang bersangkutan. Salah satu dari pendapatan daerah yaitu pajak [UND - 04].

2.2.2. Pajak Daerah

Menurut undang – undang republik indonesia nomor 28 tahun 2009 tentang pajak daerah dan retribusi daerah pasal 1 ayat 10, pajak daerah, yang selanjutnya disebut pajak, adalah kontribusi wajib kepada daerah yang terutang oleh orang pribadi atau badan yang bersifat memaksa berdasarkan undang-undang, dengan tidak mendapatkan imbalan secara langsung dan digunakan untuk keperluan daerah bagi sebesar-besarnya kemakmuran rakyat [UND - 04].

2.2.3. Jenis Pajak

Jenis pajak dibagi menjadi dua yaitu jenis pajak provinsi dan jenis pajak kabupaten/kota dan yang akan dibahas dalam penelitian ini adalah pajak kabupaten/kota [UND - 04]:

- a. Pajak Hotel
- b. Pajak Restoran
- c. Pajak Hiburan
- d. Pajak Reklame
- e. Pajak Penerangan Jalan
- f. Pajak Mineral Bukan Logam dan Batuan
- g. Pajak Parkir
- h. Pajak Air Tanah
- i. Pajak Sarang Burung Walet
- j. Pajak Bumi dan Bangunan Perdesaan dan Perkotaan.

- k. Bea Perolehan Hak atas Tanah dan Bangunan.

2.2.4. Pajak Hotel

Pajak hotel adalah pajak atas pelayanan yang disediakan oleh hotel. Objek pajak hotel adalah pelayanan yang disediakan oleh hotel dengan pembayaran, termasuk jasa penunjang sebagai kelengkapan hotel yang sifatnya memberikan kemudahan dan kenyamanan seperti fasilitas telepon, faksimile, teleks, internet, fotokopi, pelayanan cuci, seterika, transportasi, dan fasilitas sejenis lainnya yang disediakan atau dikelola hotel, termasuk fasilitas olahraga dan hiburan [UND - 04].

2.2.5. Surat Pemberitahuan Pajak Daerah

Surat pemberitahuan pajak daerah yang selanjutnya disebut SPTPD adalah surat yang oleh wajib pajak digunakan untuk melaporkan penghitungan dan/atau pembayaran pajak, objek pajak dan/atau bukan objek pajak, dan/atau harta dan kewajiban sesuai dengan ketentuan peraturan perundang-undangan perpajakan daerah [UND - 04].

2.2.6. Dasar Penagihan Pajak

Penagihan tidak dilakukan secara sembarangan oleh petugas pajak, penagihan tersebut dilakukan dengan menggunakan dasar yang sesuai. Hal ini bertujuan agar pelaksanaan penagihan dapat berjalan lancar sesuai prosedur. Dasar-dasar penagihan pajak menurut PERDA NOMOR 16 TENTANG PAJAK DAERAH adalah [UND - 04]:

Pasal 7

- (1) Tarif Pajak Hotel sebagaimana dimaksud dalam Pasal 4 ayat (3) huruf a, huruf b, huruf c, dan huruf e, ditetapkan sebesar 10% (sepuluh persen).
- (2) Tarif Pajak Hotel sebagaimana dimaksud dalam Pasal 4 ayat (3) huruf d, ditetapkan sebesar 5% (lima persen).

Pasal 8

Besarnya pokok Pajak Hotel yang terutang dihitung dengan cara mengalikan tarif sebagaimana dimaksud dalam Pasal 7, dengan dasar pengenaan pajak sebagaimana dimaksud dalam Pasal 6.

Bagian Ketiga

Masa Pajak, Penetapan dan Saat Pajak Terutang

Pasal 9

Masa Pajak Hotel adalah jangka waktu yang lamanya 1 (satu) bulan kalender.

Pasal 10

- (1) Setiap Wajib Pajak Hotel wajib mengisi SPTPD.
- (2) SPTPD sebagaimana dimaksud pada ayat (1), harus diisi dengan jelas, benar dan lengkap serta ditandatangani oleh Wajib Pajak atau kuasanya.
- (3) SPTPD sebagaimana dimaksud pada ayat (1), harus disampaikan kepada Kepala Daerah atau Pejabat yang ditunjuk selambat-lambatnya 10 (sepuluh) hari setelah berakhirnya masa pajak.
- (4) Bentuk, isi dan tata cara pengisian SPTPD akan diatur lebih lanjut dengan Peraturan Kepala Daerah.

Pasal 11

Pajak terutang dalam masa pajak, terjadi pada saat pembayaran kepada pengusaha hotel atau sejak diterbitkan SPTPD.

2.2.7. Sistem Informasi Manajemen

Sistem informasi manajemen (*Management Information System* atau sering dikenal dengan singkatannya MIS) merupakan penerapan sistem informasi di dalam organisasi untuk mendukung informasi-informasi yang dibutuhkan oleh semua tingkatan manajemen [WIR - 07].

SIM (sistem informasi manajemen) dapat didefinisikan sebagai kumpulan dari interaksi sistem-sistem informasi yang bertanggung jawab mengumpulkan

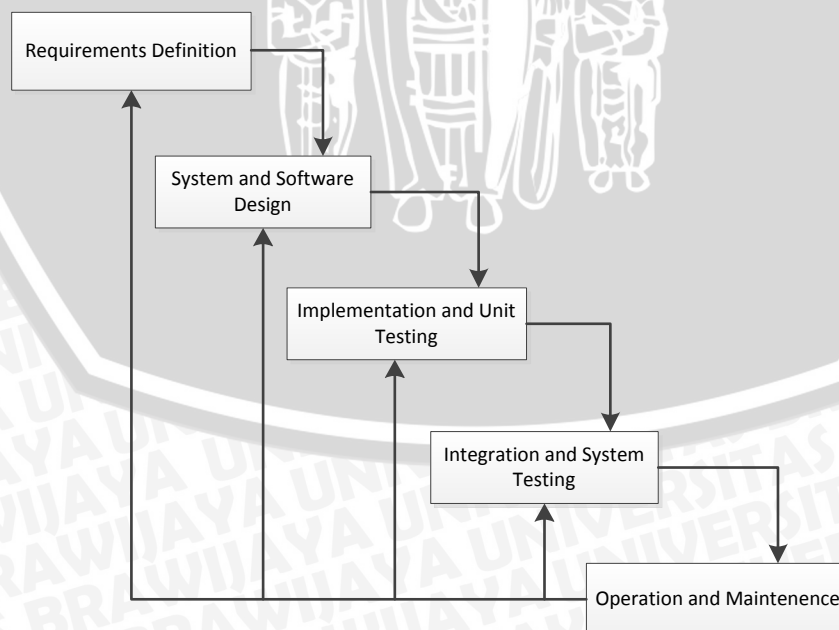
dan mengolah data untuk menyediakan informasi yang berguna untuk semua tingkatan manajemen di dalam kegiatan perencanaan dan pengendalian [WIR - 07].

2.2.8. SDLC

SDLC atau *Software Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model – model dan metodologi yang digunakan orang untuk mengembangkan sistem – sistem perangkat lunak sebelumnya (berdasarkan best practice atau cara-cara yang sudah teruji baik). SDLC memiliki beberapa model dalam penerapan tahapan proses-prosesnya. Dan model yang akan digunakan dalam penelitian ini yaitu model *waterfall* [SOM - 12].

2.2.3.1. Model Waterfall

Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan seterusnya. Berikut adalah gambar model air terjun [SOM - 12]:



Gambar 2.17 Alur *Waterfall* [SOM - 12]

a. *Requirements Analysis and Definitioin*

Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan, kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang baik.

b. *System and Software Design*

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

c. *Implementation and Unit Testing*

Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji baik secara unit.

d. *Integration and System Testing*

Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).

e. *Operation and Maintenance*

Pengujian fokus pada perangkat lunak secara dari segi logic dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan diinginkan.

2.2.9. UML

UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML hanya berfungsi untuk melakukan pemodelan. jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek [ROS - 13].



2.2.9.1. Use Case Diagram

Use case atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi - fungsi itu. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut *actor* dan *use case* [ROS - 13].


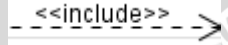
1. *Actor* merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari *actor* adalah gambar orang, tapi *actor* belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau *actor*.

Berikut adalah simbol – simbol yang ada pada diagram *use case* :

Tabel 2.1 Simbol – Simbol *Use Case*

| Simbol | Deskripsi |
|---|---|
| <p><i>Use Case</i></p>  | <p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i>; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p> |
| <p>Aktor / <i>actor</i></p>  | <p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari <i>actor</i> adalah gambar orang, tapi <i>actor</i> belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p> |

Tabel 2.1 Simbol – Simbol *Use Case* (Lanjutan)

| Simbol | Deskripsi |
|---|---|
| Asosiasi / <i>association</i>  | Komunikasi antara actor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan actor. |
| Ekstensi / <i>extend</i>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. |
| Generalisasi / <i>generalisation</i>  | Hubungan Generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| Menggunakan / <i>include</i> / <i>uses</i>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini. |

2.2.9.2. Activity Diagram







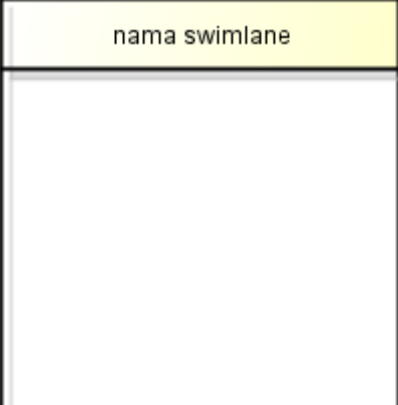
Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan *actor*, jadi aktivitas yang dapat dilakukan oleh sistem [ROS - 13].

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut [ROS - 13]:

- Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

- d. Rancangan menu yang ditampilkan pada perangkat lunak.
Berikut adalah simbol-simbol yang ada pada diagram aktivitas:






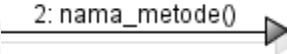
Tabel 2.2 Simbol – Simbol *Activity Diagram*

| Simbol | Deskripsi |
|--|--|
| Status awal  | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas  | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan  | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan / join  | Asosiasi penggabungan dimana jika ada pilihan aktivitas lebih dari satu. |
| Generaliasi / generalitation  | Hubungan Generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| Status akhir  | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |
| Swimlane  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |


2.2.9.3. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek - objek yang terlibat dalam sebuah *use case* beserta metode - metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case* [ROS - 13]. Berikut adalah simbol - simbol yang ada pada diagram sekuen.

Tabel 2.3 Simbol – Simbol Sequence Diagram

| Simbol | Deskripsi |
|---|--|
| <p>Actor</p>  | <p>Actor adalah pengguna sistem. pengguna dapat berarti manusia, mesin, atau sistem lain bahkan subsistem dalam sistem. Apa pun yang berinteraksi dengan sistem dari luar atau sistem batas disebut Actor.</p> |
| <p>Garis hidup / lifetime</p>  | <p>Menyatakan kehidupan suatu objek</p> |
| <p>Boundary</p>  | <p>Boundary biasanya berupa tepi dari sistem, seperti user interface, atau suatu alat yang berinteraksi dengan sistem lain.</p> |
| <p>Control</p>  | <p>Control element mengatur aliran dari informasi untuk sebuah scenario. Perilaku dan perilaku bisnis umumnya diatur oleh objek ini.</p> |
| <p>Entity</p>  | <p>Entity biasanya elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau model object.</p> |
| <p>Pesan Tipe Call</p>  | <p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.</p> |

Tabel 2.3 Simbol – Simbol *Sequence Diagram* (Lanjutan)

| Simbol | Deskripsi |
|---|---|
| Pesan Tipe <i>return</i>  | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |

2.2.9.4. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [ROS - 13].

- a. Atribut merupakan variabel – variabel yang dimiliki oleh suatu kelas
 - b. Operasi atau metode adalah fungsi – fungsi yang dimiliki oleh suatu kelas
- Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak atau *programmer* dapat membuat kelas-kelas di dalam program perangkat lunak sesuai dengan perancangan diagram kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas tersebut.

- a. Kelas main
Kelas yang memiliki fungsi awal dieksekusi ketika system dijalankan.
- b. Kelas yang menangani tampilan sistem (*view*)
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
- c. Kelas yang diambil dari pendefinisian *use case* (*controller*)
Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian usecase, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak
- d. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Semua tabel yang dibuat di basis data dapat dijadikan kelas, namun untuk tabel dari hasil relasi atau atribut multivalued pada ERD dapat dijadikan kelas tersendiri dapat juga tidak

asalkan pengaksesannya dapat dipertanggung jawabkan atau tetap ada di dalam perancangan kelas.

2.2.10. Pengujian

Pengujian perangkat lunak merupakan salah satu tahap dalam rekayasa perangkat lunak. Pengujian mempunyai sasaran mendeteksi adanya kesalahan yang terjadi pada perangkat lunak yang sedang dibangun, sehingga pengujian yang baik adalah pengujian yang dapat mendeteksi adanya kesalahan. Secara garis besar, pengujian dapat dikategorikan ke dalam 2 kelompok pengujian, yaitu *black box* dan *white box*. Pengujian yang di gunakan dalam penelitian ini adalah pengujian *black box* [KOM - 14].

2.2.10.1. Pengujian *Black Box*

Dengan mengetahui spesifikasi fungsi dari produk, pengujian dilakukan untuk menunjukkan bahwa setiap fungsi beroperasi sepenuhnya. Pada umumnya, pengujian kotak hitam lebih sering digunakan dalam rekayasa perangkat lunak yang tidak terlalu kompleks, dengan alasan lebih mudah dilakukan. Pengujian ini digunakan untuk menunjukkan bahwa fungsi perangkat lunak telah beroperasi sesuai dengan yang diinginkan, tanpa memperhatikan struktur logika internal dari perangkat lunak, sehingga kemungkinan ada kesalahan internal yang tidak terdeteksi dengan pengujian kotak hitam ini. Dan salah satu tipe pengujian *black box* yang digunakan dalam penelitian ini adalah *User Acceptance Testing* (UAT) [KOM - 14].

2.2.10.1.1. *Functional Testing*

Functional Testing merupakan tes yang dilakukan untuk mengetahui apakah sistem yang di tes sudah berjalan sesuai dengan rekuirmen sistem tersebut tanpa harus mengetahui struktur internal dari sistem dan tidak mengetahui bagaimana sistem bekerja [KOM - 14].

2.2.10.1.2. User Acceptance Testing (UAT)

User Acceptance Testing (UAT) adalah proses untuk mendapatkan konfirmasi bahwa sebuah sistem memenuhi yang disepakati persyaratan. Sebuah *Subject Matter Expert* (SME), lebih baik pemilik atau klien dari benda yang diuji, memberikan konfirmasi tersebut setelah pengadilan atau diperiksa. Dalam pengembangan perangkat lunak, UAT adalah salah satu tahap akhir proyek dan sering terjadi sebelum klien atau pelanggan menerima sistem baru. Dalam UAT dapat dinilai menggunakan dua persepsi yaitu *Perceived Usefulness*, dan *Perceived Ease of Use* [KOM - 14].

1. *Perceived Usefulness*

Perceived usefulness didefinisikan di sini sebagai sejauh mana seseorang percaya bahwa menggunakan sistem tertentu akan meningkatkan prestasi kerja mereka dan berguna bagi suatu organisasi dalam memudahkan penyelesaian pekerjaan [DAV - 89].

1. *Perceived Ease of Use*

Perceived ease of use, berarti sebaliknya yaitu mengacu pada sejauh mana seseorang percaya bahwa menggunakan sistem yang akan digunakan tersebut mudah untuk dipelajari dan digunakan [DAV - 89].