

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Secara sederhana sistem adalah sekelompok komponen yang saling berhubungan, bekerja sama untuk mencapai tujuan bersama dengan menerima *input* serta menghasilkan *output* dalam transformasi yang teratur.

Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*process*), dan keluaran (*output*). Komponen-komponen dalam sistem, mencakup perangkat keras (*hardware*), perangkat lunak (*software*), prosedur-prosedur, perangkat manusia (*brainware*), dan informasi (*information*). Suatu sistem seanehtiasa tidak terlepas dari lingkungan sekitarnya, maka umpan balik (*feedback*) disamping dapat berasal dari *output*, juga dapat berasal dari lingkungan sistem tersebut.

2.1.1. Pengertian Sistem

Sistem berasal dari bahasa Yunani, yaitu *systema*, yang artinya himpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Selain itu, sistem dapat diartikan sekelompok elemen yang independen, namun saling berkaitan sebagai suatu kesatuan.

Sistem terdiri atas struktur dan proses. Struktur sistem merupakan unsur-unsur yang membentuk sistem tersebut, sedangkan proses sistem menjelaskan cara kerja setiap unsur sistem dalam mencapai tujuan.

Menurut (E, Y Anggraeni. Irviani, 2017) “Sistem adalah kumpulan orang yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan

terstruktur dan membentuk satu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan”.

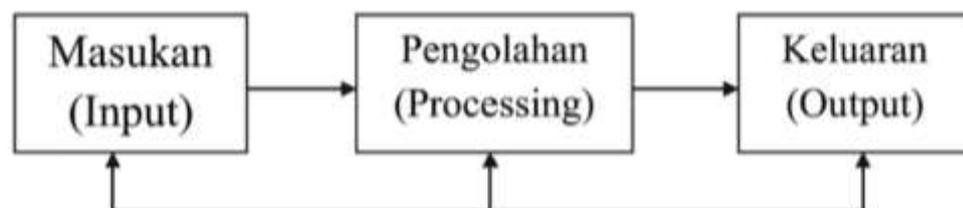
Sedangkan (Hamidin, 2017) menyimpulkan bahwa:

Sistem didefinisikan sebagai kumpulan dari elemen-elemen berupa data, jaringan kerja dari prosedur-prosedur yang saling berhubungan, sumber daya manusia, teknologi baik *hardware* maupun *software* yang saling berinteraksi sebagai satu kesatuan untuk mencapai tujuan atau sasaran tertentu yang sama.

Begitu juga menurut Schronderberg yang ditulis dalam (Muslihudin.

Oktafianto., 2016) Secara ringkas menjelaskan bahwa sistem adalah:

1. Komponen-komponen yang saling berhubungan satu sama lain.
2. Suatu keseluruhan tanpa memisahkan komponen pembentuknya.
3. Bersama-sama dalam mencapai tujuan.
4. Memiliki *input* dan *output*-nya yang dibutuhkan oleh sistem lainnya.
5. Terdapat proses yang mengubah *input* menjadi *output*.
6. Menunjukkan adanya entropi.
7. Memiliki aturan.
8. Memiliki subsistem yang lebih kecil.
9. Memiliki deferensi antar subsistem.
10. Memiliki tujuan yang sama meskipun memulainya berbeda.



Sumber: (Muslihudin. Oktafianto., 2016)

Gambar II.1
Model Sistem

Berdasarkan pengertian diatas, maka dapat disimpulkan bahwa sistem merupakan sekumpulan komponen yang saling berhubungan dan bekerja sama sehingga membentuk jaringan kerja yang memiliki *input* dan *output* lalu di proses yang melaksanakan suatu fungsi untuk mencapai suatu tujuan tertentu atau tujuan yang telah ditentukan dengan sistematis dan terstruktur.

2.1.2. Pengertian Informasi

Informasi adalah data yang diproses menjadi suatu bentuk yang lebih berguna dan berarti bagi yang menerimanya dalam aktivitas pembuatan keputusan. Menurut (Mulyani, 2017) “Informasi merupakan data yang sudah diolah yang ditunjukkan untuk seseorang, organisasi, ataupun siapa saja yang membutuhkan”.

Sedangkan menurut (D, Yulia. Pratita, 2015) “Informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi penerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan”.

Begitu juga menurut (Muslihudin. Oktafianto., 2016) “Informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan”. Informasi berguna untuk pembuat keputusan karena informasi menurunkan ketidakpastian (meningkatkan pengetahuan). Informasi menjadi penting karena berdasarkan informasi para pengelola dapat mengetahui kondisi obyektif perusahaannya. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan metode atau cara-cara tertentu.

Dengan demikian dapat disimpulkan bahwa informasi adalah data yang diproses menjadi suatu bentuk yang lebih berguna dan berarti bagi yang menerimanya dalam aktivitas pembuatan keputusan yang dikumpulkan dengan metode atau cara-cara tertentu.

2.1.3. Pengertian Sistem Informasi

Pada perusahaan atau instansi yang mengikuti perkembangan teknologi dan informasi, sistem informasi dipakai sebagai alat terapan dan kombinasi antara teknologi dengan sistem yang dipakai dalam perusahaan atau instansi tersebut.

Menurut (Aswati, Mulyani, Siagian, & Zikra Syah, 2015) “Sistem informasi merupakan seperangkat fungsi operasional manajemen kepada yang mampu menghasilkan suatu keputusan yang tepat, cepat dan jelas sehingga menjadi suatu susunan yang disusun secara sistematis dan teratur”.

Sedangkan menurut (Hutahaean, 2015) “Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang dibutuhkan”.

Dengan demikian dapat disimpulkan bahwa sistem informasi adalah suatu serangkaian prosedur pengumpulan data yang terdiri dari komponen-komponen yang terkait dengan *hardware*, *software* dan sumber daya manusia sehingga dapat menghasilkan informasi untuk mencapai suatu tujuan.

2.1.4. Pengertian Sistem Informasi Akuntansi

Sistem informasi akuntansi merupakan sistem pengolahan data akuntansi yang terdiri dari koordinasi manusia, alat dan metode yang berinteraksi dalam suatu wadah organisasi yang berstruktur untuk menghasilkan informasi akuntansi keuangan dan informasi akuntansi manajemen berstruktur.

Menurut Krismiaji dalam (Lumanaw & Tinangon, 2016) “Sistem informasi akuntansi adalah sebuah sistem yang memproses data dan transaksi guna menghasilkan informasi yang bermanfaat untuk merencanakan, mengendalikan dan mengoperasikan bisnis”. Sedangkan menurut George H. Bodnar dan William S Hopwood dalam (Fauzi, 2017) “Sistem Informasi Akuntansi adalah kumpulan

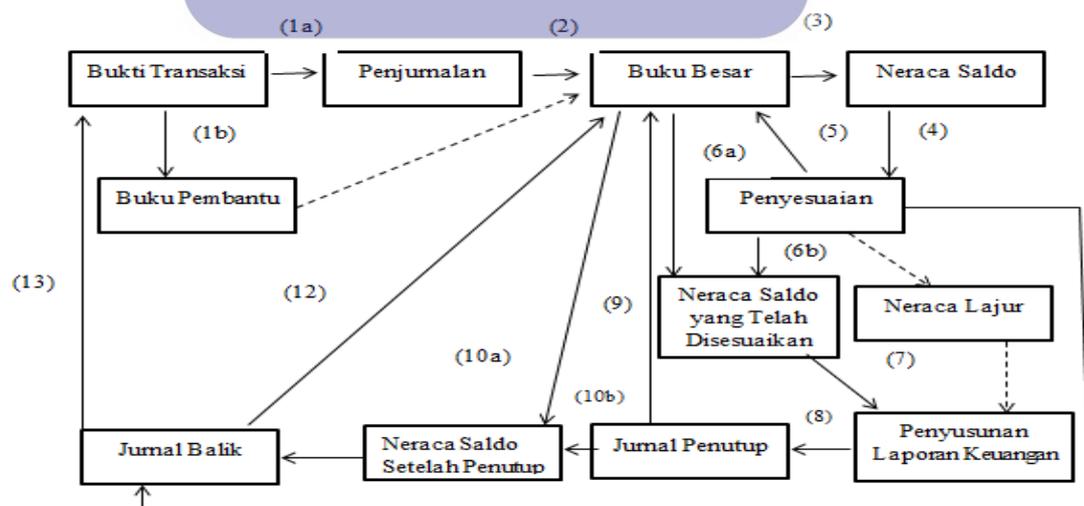
sumber daya seperti manusia dan peralatan yang diatur untuk mengubah data ekonomi menjadi informasi yang berguna”.

Bedasarkan pengertian sistem informasi akuntansi yang telah dijelaskan, maka dapat disimpulkan bahwa sistem informasi akuntansi adalah sekumpulan komponen-komponen yang terintegrasi dan saling bekerja sama untuk mengolah data-data keuangan menjadi sebuah informasi yang dapat digunakan untuk mengambil keputusan oleh para pemegang kepentingan.

2.1.5. Pengertian Siklus Akuntansi

Siklus akuntansi adalah suatu proses penyediaan laporan keuangan perusahaan atau instansi untuk suatu periode waktu tertentu. Informasi berupa laporan keuangan dihasilkan melalui proses akuntansi yang panjang. Pada proses tersebut terdapat tahapan yang harus dipenuhi untuk mendapatkan hasil laporan yang baik, valid dan akuntabel.

Menurut (Bahri, 2016) “Siklus akuntansi adalah tahapan-tahapan mulai dari terjadinya transaksi sampai dengan penyusunan laporan keuangan sehingga siap untuk pencatatan berikutnya”.



Sumber: (Bahri, 2016)

Gambar II.2
Siklus Akuntansi

Menurut (Pujiyanti, 2015) “Siklus Akuntansi merupakan proses penyusunan suatu laporan keuangan yang dapat dipertanggungjawabkan dan serta diterima secara umum prinsip-prinsip dan kaidah akuntansi, prosedur-prosedur, metode-metode, serta teknik-teknik dari segala sesuatu yang dicakup dalam ruang akuntansi dalam suatu periode tertentu.”.

Dengan penjelasan diatas dapat disimpulkan bahwa siklus akuntansi adalah suatu kegiatan yang dilakukan terus menerus yang didalamnya terdapat berbagai proses, dimulai dari proses transaksi hingga penyusunan laporan keuangan untuk persiapan aktivitas periode akuntansi selanjutnya dengan pembuatan jurnal balik.

2.1.6. Pengertian Penggajian

Sumber daya manusia yang profesional diperlukan untuk mencapai tujuan perusahaan atau instansi dalam mengendalikan perusahaan atau instansi tersebut. Sebagai imbalan kepada sumber daya manusia tersebut, perusahaan atau instansi harus menjanjikan suatu kontra prestasi dalam bentuk gaji sebagai penghargaan usaha tenaga kerja dengan jumlah yang telah ditentukan pada periode waktu tertentu.

Menurut Sujarweni dalam (Jiwandono, 2017) “Gaji adalah pembayaran atas jasa-jasa yang dilakukan oleh karyawan yang dilakukan perusahaan setiap bulan”. Sedangkan menurut Mulyadi dalam (Aqurisma & Nurhayati, 2017) “Gaji umumnya merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan yang mempunyai jenjang jabatan manajer, sedangkan upah umumnya merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan pelaksana (buruh)”.

Dengan demikian dapat disimpulkan bahwa penggajian adalah balas jasa yang diberikan oleh perusahaan atau instansi dalam bentuk pembayaran sebagai balasan atas seseorang karena telah melaksanakan tugas atau jasa yang berguna untuk mencapai tujuan perusahaan atau instansi yang bersangkutan.

2.1.7. Pengertian Penjurnalan

Jurnal adalah rincian semua transaksi keuangan dan akun-akun yang mempengaruhi transaksi. Pencatatan semua transaksi keuangan suatu perusahaan atau instansi yang dicatat secara sistematis dan bertujuan untuk pendataan, termasuk di dalamnya jumlah transaksi, nama-nama transaksi baik memengaruhi atau dipengaruhi dan waktu transaksi berjalan.

Menurut (Bahri, 2016) “Jurnal adalah pencatatan yang sistematis dan kronologis atas transaksi yang terjadi pada perusahaan”. Bahri menjelaskan jurnal merupakan langkah awal dalam siklus akuntansi. Jurnal adalah catatan berupa pendebitan dan pengkreditan atas pengaruh transaksi dan disertai penjelasan-penjelasan yang diperlukan untuk transaksi tersebut.

Adapun jurnal penggajian yang merupakan catatan beban gaji yang telah dikeluarkan perusahaan. Jurnal untuk pencatatan penggajian tersebut adalah sebagai berikut:

Pencatatan jurnal gaji:

Biaya Gaji	xxx
Kas	xxx

2.1.8. Metode Pengembangan *Waterfall*

Nama model *Waterfall* sebenarnya adalah *Linear Sequential Model*. Model ini sering disebut dengan *classic life cycle* atau model *waterfall*. *Waterfall* pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *Software Engineering* (SE). *Waterfall* melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Menurut (Shalahuddin & Rosa, 2015) “Model *waterfall* adalah model SDLC yang paling sederhana”. Shalahuddin & Rosa juga menjelaskan Model *waterfall* hanya cocok untuk pengembangan perangkat lunak dengan spesifikasi yang tidak berubah-ubah. Model air terjun ini menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian dan tahap pendukung (*support*).



Sumber: (Shalahuddin & Rosa, 2015)

Gambar II.3
Alur Model Waterfall

Adapun tahapan dari model *waterfall* menurut (Shalahuddin & Rosa, 2015) yaitu:

1. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara insentif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

2. Desain

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 pengkodean. 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.

3. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sistem sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

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

5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.9. Pengertian *Database*

Basis data adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu file terdapat beberapa *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam. Satu *record* terdiri dari *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang

lengkap dan direkam dalam satu *record*. Suatu basis data berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut.

Menurut Lubis dalam (Anna, Nurmalasari, & Yusnita, 2018) “Basis data adalah tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (organisasi/perusahaan) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut”.

Kebutuhan basis data sangat dibutuhkan dalam sistem informasi dengan menggunakan *Database Management System* (DBMS) dan bahasa *MySQL*. Adapun *Database Management System* (DBMS) menurut (Shalahuddin & Rosa, 2015) “*Database Management System* (DBMS) adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data”.

1. MySQL

MySQL adalah program *database* yang mampu mengirim dan menerima data dengan sangat cepat dan multi *user*. MySQL memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. MySQL *free software* bebas menggunakan *database* ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada dibawah *General Public Lisensi* (GPL).

Menurut (Lubis & Adrian, 2017) “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user* dengan sekitar 6 juta instalasi di seluruh dunia”. Beberapa kelebihan MySQL adalah bebas *download*, stabil dan tangguh, fleksibel dengan berbagai pemrograman serta perkembangan *Software* yang cukup erat.

2. Java

Java adalah suatu bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, *desktop*, *web*, *mobile* dan lainnya, sebagaimana dibuat dengan

menggunakan bahasa pemrograman konvensional yang lain. Bahasa pemrograman *Java* ini berorientasi objek *Object Oriented Programming* (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi.

Menurut (Nofriadi, 2018) “Bahasa pemrograman *java* merupakan salah satu dari sekian banyak bahasa pemrograman yang dapat dijalankan diberbagai sistem operasi termasuk telepon genggam”.

3. *Netbeans* IDE

Netbeans IDE adalah suatu aplikasi untuk membuat suatu program dengan bahasa pemrograman *java* maupun bahasa pemrograman yang lain. Dengan mudahnya penggunaan aplikasi *netbeans* ini maka *programmer* sering menggunakan aplikasi ini sebagai wadah untuk membuat berbagai macam program yang bersifat *Open Source*.

Menurut (Nofriadi, 2018) “*Netbeans* merupakan sebuah aplikasi *Integrated Development Environment* (IDE) yang berbasiskan *java* dari *Sun Microsystem* yang berjalan diatas *Swing* dan banyak digunakan sekarang sebagai editor untuk berbagai bahasa pemrograman”.

2.2. Peralatan Pendukung (*Tools System*)

Peralatan pendukung merupakan suatu penjelasan mengenai peralatan-peralatan pendukung yang dipakai dalam membuat suatu sistem usulan. Media pendukung sistem merupakan alat yang dapat digunakan untuk menggambarkan bentuk logika model dari suatu sistem, dimana simbol-simbol, lambang-lambang dan diagram-diagram menunjukkan secara tepat arti fisiknya.

2.2.1. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) merupakan pengganti dari metode analisis berorientasi objek dan desain berorientasi objek (OOA&D) yang

dimunculkan sekitar akhir tahun 90-an. UML merupakan gabungan dari metode *Booch*, *Rumbaugh* (OMT) dan *Jacobson*. Tetapi UML ini akan mencakup lebih luas dari pada OOA&D. Pada pertengahan pengembangan UML dilakukan standarisasi proses dengan *Object Management Group* (OMG) dengan harapan UML akan menjadi bahasa standard pemodelan pada masa yang akan datang.

Menurut (Shalahuddin, 2016:133) UML (*unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek”.

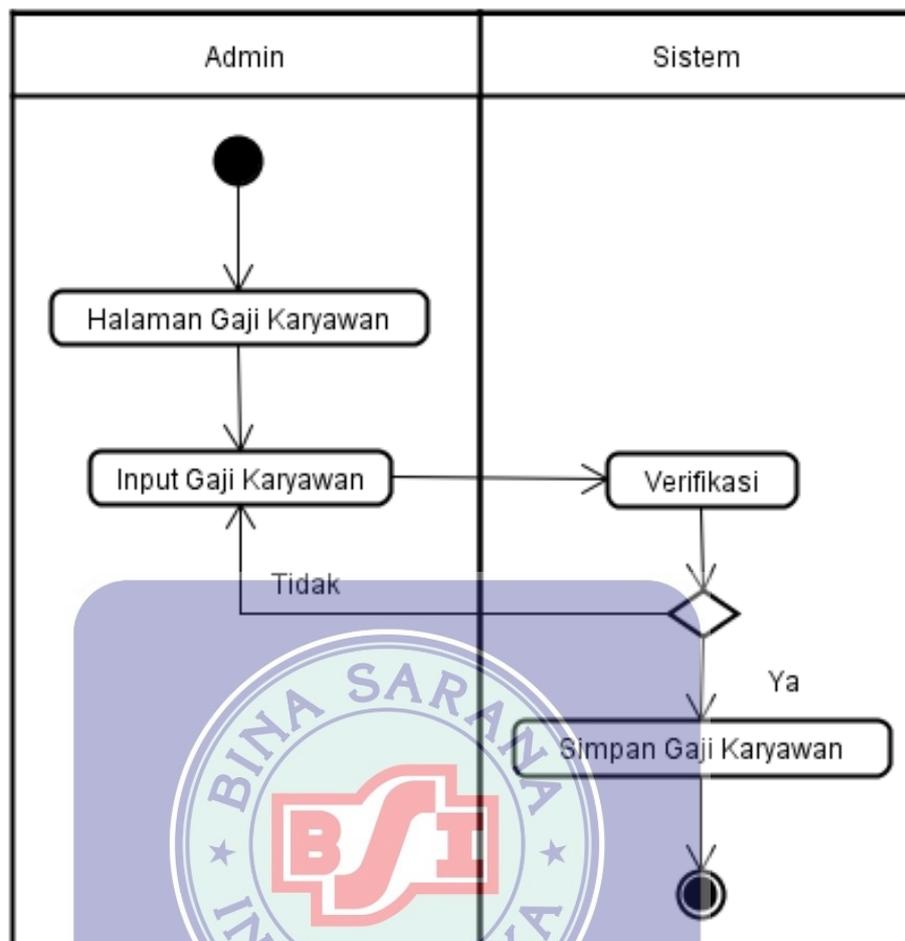
1. *Activity Diagram*

Activity diagram menggambarkan tentang aktifitas yang terjadi pada sistem dari pertama sampai akhir.

Menurut (Shalahuddin & Rosa, 2015) “Diagram aktivitas atau *activity diagram* menggambarkan aliran kerja (*workflow*) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
- d. Rancangan menu yang ditampilkan pada perangkat lunak.



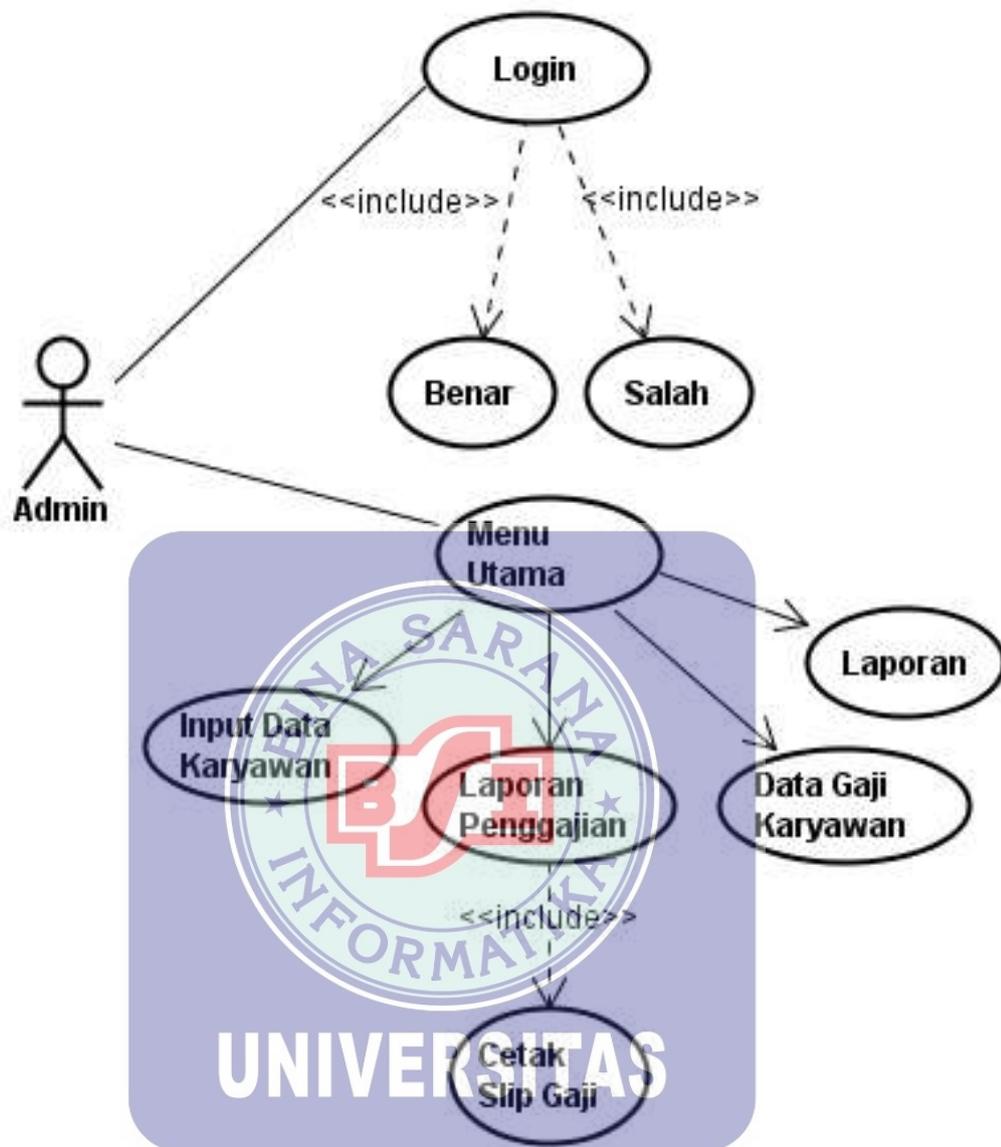
Sumber: (Moenir & Yuliyanto, 2017)

Gambar II.4
Contoh Activity Diagram Sistem Pembayaran Gaji

2. UseCase Diagram

Use case diagram digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa yang dilakukannya.

Menurut (Shalahuddin & Rosa, 2015) “Use case diagram atau diagram use case merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor 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.

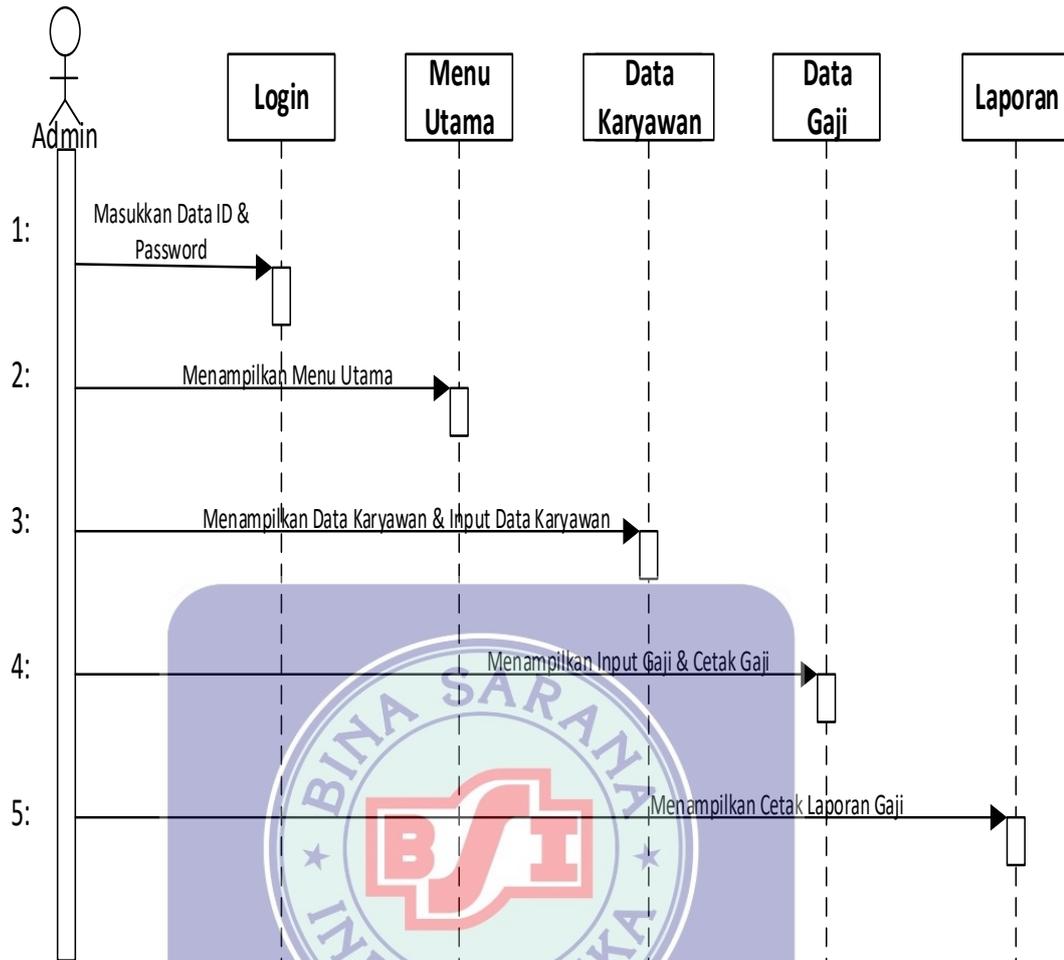


Sumber: (Widarma & Rahayu, 2017)

Gambar II.5
Contoh Usecase Diagram Sistem Pembayaran Gaji

3. Sequence Diagram

Sequence Diagram adalah sebuah diagram yang menggambarkan interaksi antar objek di dalam sebuah sistem. Interaksi tersebut berupa *message* yang digambarkan terhadap waktu. Menurut (Shalahuddin & Rosa, 2015) “Diagram Sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”.

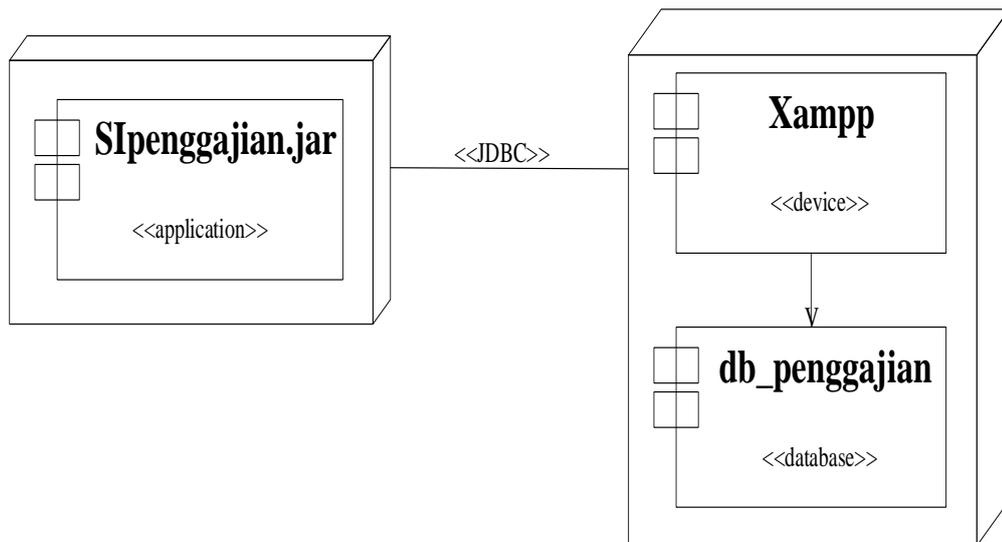


Sumber: (Widarma & Rahayu, 2017)

Gambar II.6
Contoh Sequence Diagram Sistem Pembayaran Gaji

4. Deployment Diagram

Diagram deployment atau *deployment diagram* yaitu salah satu diagram pada UML yang menunjukkan tata letak suatu sistem secara fisik, dapat juga dikatakan untuk menampilkan bagian-bagian *software* yang terdapat pada *hardware* dan digunakan untuk menerapkan suatu sistem dan hubungan antara komponen *hardware*. Jadi *deployment diagram* intinya untuk menunjukkan letak *software* pada *hardware* yang digunakan sistem. Menurut (Shalahuddin & Rosa, 2015) “*Deployment Diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi”.



Sumber: (Mintarsih, 2015)

Gambar II.7
Contoh Deployment Diagram Sistem Pembayaran Gaji

Berdasarkan pengertian, dapat disimpulkan bahwa UML merupakan bahasa yang mempermudah pembaca dalam memahami cara berjalannya sebuah sistem perangkat lunak dengan menggunakan sebuah gambar.

2.2.2. Pengertian *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah model teknik pendekatan yang menyatakan atau menggambarkan hubungan suatu model.

Menurut (Shalahuddin & Rosa, 2015) "*Entity Relationship Diagram* (ERD) adalah pemodelan awal basis data yang paling banyak digunakan". *Entity Relationship Diagram* (ERD) merupakan suatu diagram yang menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa data-data tersebut terdiri dari objek-objek dasar yang mempunyai hubungan atau relasi satu sama lain. Penggambaran *Entity Relationship Diagram* (ERD) akan membantu perancangan proses yang akan dituangkan dalam bentuk baris-baris program.

Komponen-komponen dalam *Entity Relationship Diagram* (ERD) adalah sebagai berikut:

1. Entitas (*Entity*)

Entitas merupakan data inti yang akan disimpan; bakal *table* pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.

2. Atribut

Field atau kolom data yang butuh disimpan dalam suatu entitas.

3. Atribut Kunci Primer

Filed atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses *record* yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).

4. Atribut Multinilai (*Multivalue*)

Field atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

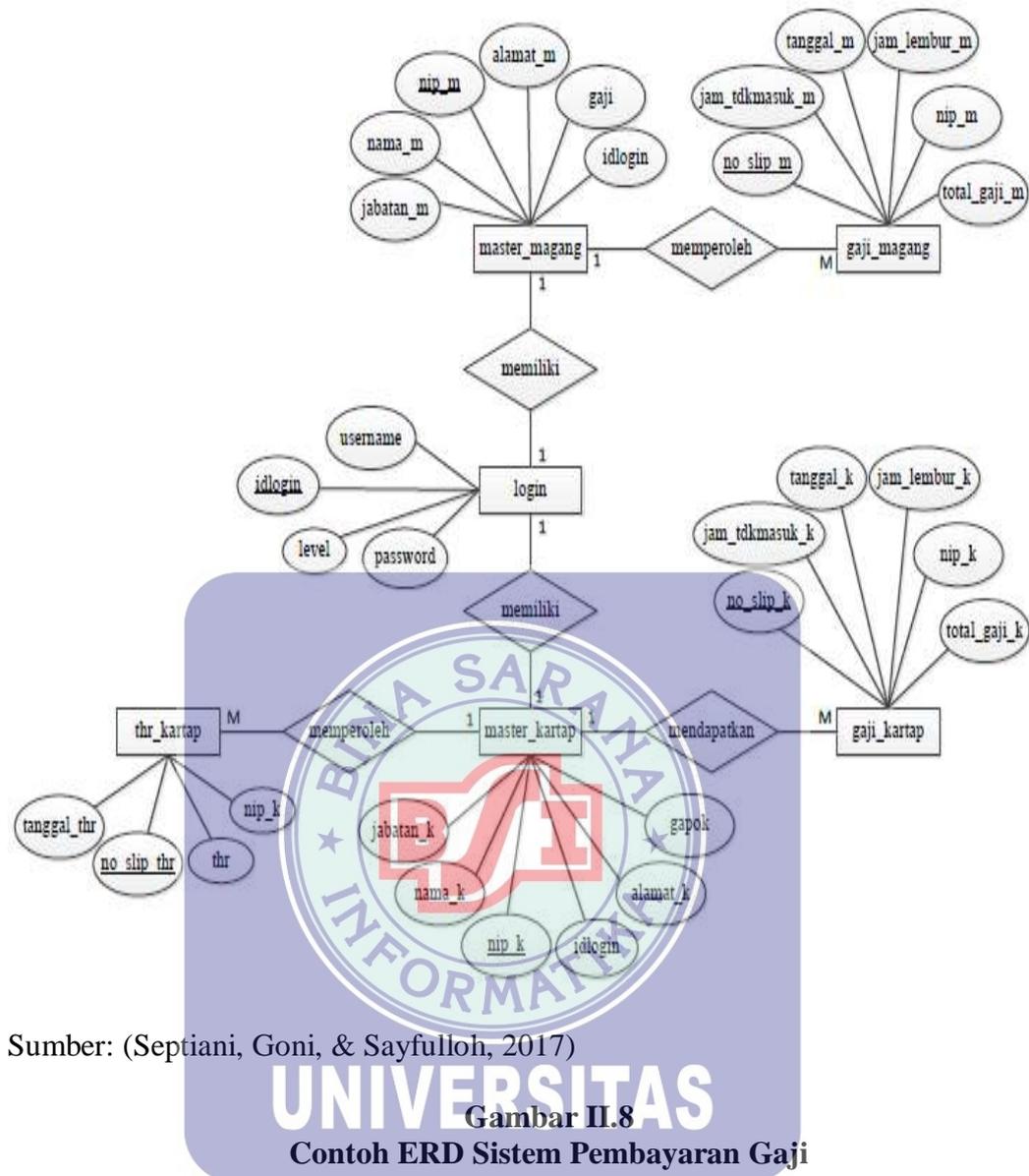
5. Relasi (*Relation*)

Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.

6. Asosiasi (*Associstion*)

Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian.

Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas 1 ke N atau sering disebut dengan *one to many* menghubungkan entitas A dan entitas B.



Sumber: (Septiani, Goni, & Sayfullloh, 2017)

Gambar II.8
Contoh ERD Sistem Pembayaran Gaji

Dari penjelasan diatas dapat disimpulkan bahwa *Entity Relationship Diagram* (ERD) pemodelan basis data dengan menggunakan diagram relasi antar entitas, untuk memodelkan struktur data dan hubungan antar data dan digambarkan dengan menggunakan beberapa notasi dan simbol.

2.2.3. Pengertian *Logical Record Structure* (LRS)

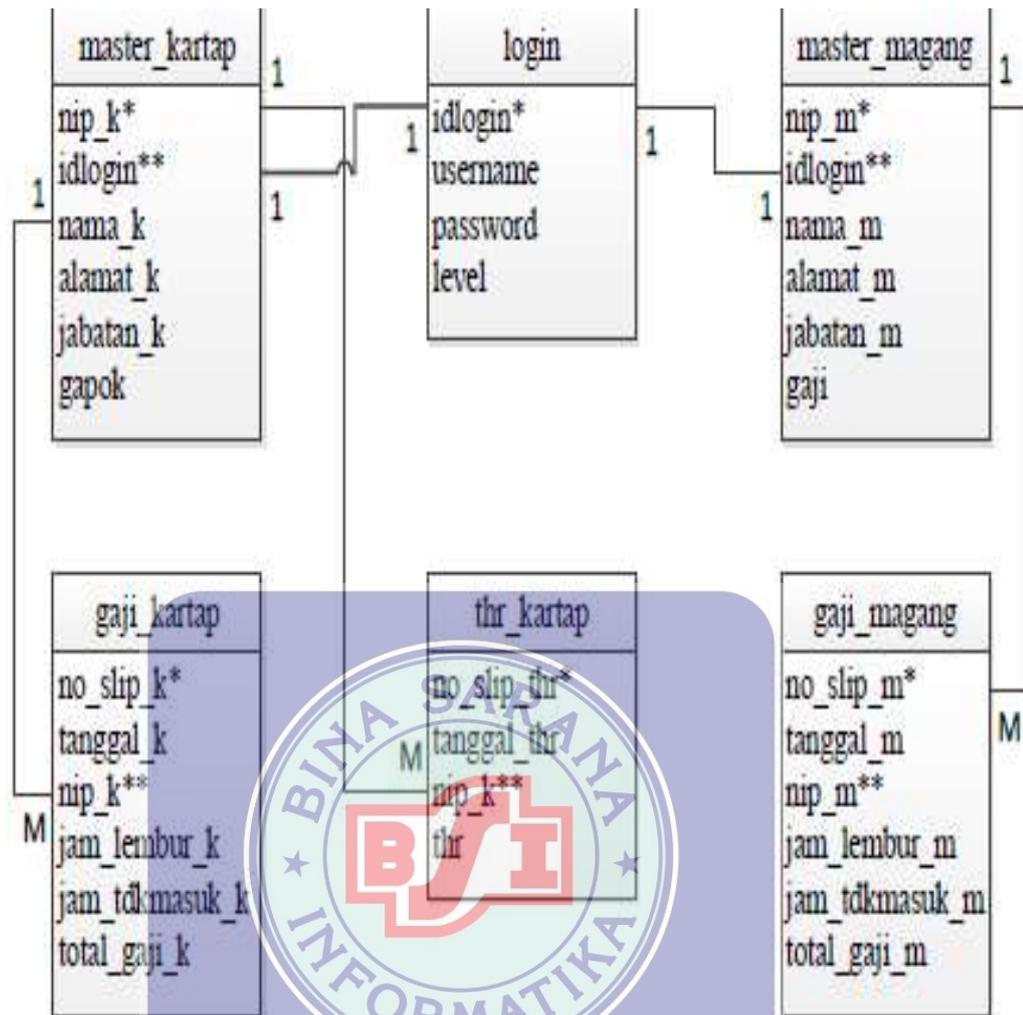
Logical Record Structured (LRS) merupakan representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil kelas antar himpunan entitas pada diagram E-R.

Menurut Simarmata dan Paryudi dalam jurnal (Fridayanthie & Mahdiati, 2016) menjelaskan bahwa: “*Logical Record Structured (LRS)* adalah representasi dari struktur *record-record* pada tabel-tabel yang terbentuk dari hasil relasi antar himpunan entitas”. Menentukan kardinalitas, jumlah tabel dan *ForeignKey (FK)*.

Sedangkan menurut Wulandari dalam (Sumirat & Jakaria, 2018) “*Logical Record Structured (LRS)* dibentuk dengan nomor dari tipe *record*. Beberapa tipe *record* digambarkan oleh kotak empat persegi panjang dan dengan nama yang unik”. Sumirat & Jakaria menjelaskan *Logical Record Structured (LRS)* terdiri dari *link-link* diantara tipe *record*. *Link* ini menunjukkan arah dari satu tipe *record* lainnya. Banyak *link* dari LRS yang diberi tanda *field-field* yang kelihatan pada kedua *link* tipe *record*. Dua metode yang dapat digunakan dalam menggambar LRS yakni dengan hubungan kedua model yang dapat dikonversikan ke LRS atau dengan hubungan *Entity Relationship Diagram (ERD)* dan langsung dikonversikan ke LRS.

Berikut adalah cara membentuk skema *database* atau *Logical Record Structured (LRS)* berdasarkan *Entity Relationship Diagram* :

1. Jika relasinya satu-ke-satu maka *foreign key* diletakan pada salah satu dari dua entitas yang ada atau menyatukan kedua entitas tersebut.
2. Jika relasinya satu-ke-banyak, maka *foreign key* diletakan pada entitas *Many*
3. Jika relasinya banyak-ke-banyak, maka dibuat file konektor yang berisi dua *foreign key* yang berasal dari kedua entitas.



Sumber: (Septiani et al., 2017)

UNIVERSITAS

Gambar II.9
Contoh LRS Sistem Pembayaran Gaji

Dari penjelasan diatas dapat disimpulkan bahwa *Logical Record Structured* (LRS merupakan suatu *record* dari tabel yang ada pada *database* dan direlasikan untuk mempermudah logika dari suatu program yang dibuat.

2.2.5. Blackbox Testing

Metode *Blackbox testing* memfokuskan pada keperluan fungsional dari *software*. Karena itu *Blackbox testing* memungkinkan pengembangan *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program.

Menurut Simanjutak dalam (Harahap, Sukanto, & Safriadi, 2016) “*Blackbox testing* adalah cara pengujian dilakukan dengan hanya menjalankan atau mengeksekusi unit atau modul kemudian diamati apakah hasil dari unit itu sesuai dengan yang diinginkan”. Sedangkan Menurut (Shalahuddin & Rosa, 2015) “*Blackbox testing* adalah perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”.

Shalahuddin & Rosa menjelaskan pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses *login* maka kasus uji yang dibuat adalah:

1. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.

Tabel II.1
Contoh *Blacbox Testing*

No.	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	<i>User Name</i> tidak diisi kemudian ditekan OK	<i>User Name</i> (Kosong) <i>Password</i> (Kosong)	Sistem akan menolak akses <i>user</i> dan membuat kursor berada pada posisi <i>user name</i>	Sesuai	Valid
2	<i>User Name</i> diisi dan <i>password</i> tidak diisi kemudian ditekan OK	<i>User Name</i> (Diisi) <i>Password</i> (Kosong)	Sistem akan menolak akses <i>user</i> dan membuat posisi kursor berada pada posisi <i>password</i>	Sesuai	valid
3	<i>User Name</i> diisi dengan salah dan <i>password</i> diisi dengan benar	<i>User Name</i> Coba (Salah) <i>Password</i> (Benar)	Sistem akan menolak akses <i>user</i> dan menampilkan " <i>user</i> tidak dikenal"	Sesuai	Valid
4	<i>User Name</i> diisi dengan benar dan <i>password</i> diisi dengan salah	<i>User Name</i> Coba (Benar) <i>Password</i> (Salah)	Sistem akan menolak akses <i>user</i> dan menampilkan " <i>user</i> tidak dikenal"	Sesuai	Valid
5	<i>User Name</i> diisi dengan benar dan <i>password</i> diisi dengan benar	<i>User Name</i> (Benar) <i>Password</i> (Benar)	Sistem akan menerima akses <i>login</i> dan menampilkan Menu Utama	Sesuai	Valid

Sumber: (Septiani et al., 2017)

Berdasarkan definisi di atas dapat disimpulkan bahwa "*Blackbox testing* yaitu perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian dengan metode *Blackbox testing* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program.