BAB II
TINJAUAN PUSTAKA

2.1 Pengertian Pasar

Menurut W.J. Stanton dalam buku Muhammad Abdul Halim (2018), pasar adalah kumpulan orang yang mempunyai keinginan untuk memenuhi kebutuhan, uang untuk belanja, serta kemauan untuk membelanjakannya, jadi sesuatu disebut pasar jika ada tiga syarat utama, yaitu: sekumpulan orang, daya beli (uang), dan kemauan untuk membeli barang atau jasa.

Menurut Damsar dan Indrayani (2018) dalam bahasa latin, pasar dapat ditelusuri melalui akar dari kata *mercatus*, yang bermakna berdagang atau tempat berdagang. Sedangkan pengertian pasar dalam bahasa arab yaitu *bazar* yang bermakna suatu peranata ekonomi dan sekaligus cara hidup, suatu gaya umum dari kegiatan ekonomi yang mencapai segala aspek dari masyarakat, dan suatu dunia sosial budaya yang lengkap dalam sendirinya.

2.2 Sistem

Effendi (2020) menyatakan bahwa sistem adalah bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud. Secara garis besar ada dua kelompok pendekatan sistem, yaitu Pendekatan sistem yang lebih menekankan pada elemen-elemen atau kelompoknya didefinisikan sebagai Suatu jaringan kerja dari prosedur prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu aturan tertentu.

2.3 Sewa

Menurut Kamus Besar Bahasa Indonesia, sewa adalah pemakaian sesuatu dengan membayar uang. Pengertian lain menurut Kamus Besar Bahasa Indonesia, sewa adalah uang yang dibayarkan karena memakai atau meminjam sesuatu.

Menurut Subekti Soekanto dan Sri Mamudji (2016), sewa menyewa adalah suatu perjanjian dengan mana pihak yang satu mengikatkan dirinya untuk

memberikan kepada pihak yang lainnya kenikmatan dari sesuatu barang, selama suatu waktu tertentu dengan pembayaran suatu harga yang oleh pihak yang tersebut terakhir itu disanggupi pembayarannya.

2.4 *Website*

Menurut Sidik dalam Arizona (2017) mengatakan bahwa Situs Web (*Website*) awalnya merupakan suatu layanan sajian informasi yang menggunakan konsep hiperlink yang memudahkan *surfer* (sebutan bagi pemakai komputer yang melakukan penyelusuran informasi di internet) untuk mendapatkan informasi dengan cukup mengklik suatu link berupa teks atau gambar maka informasi dari teks atau gambar akan ditampilkan secara lebih terperinci (detail).

.

2.5 Pengertian Pemrograman PHP

Menurut Firmansyah (2018) mendefinisikan bahwa PHP (*Hypertext Preprocessor*) adalah PHP mengijinkan pengembang untuk menempelkan kode di dalam HTML dengan menggunakan bahasa yang sama seperti perl dan UNIX shells. Sedangkan menurut Isty & Afifah (2018), PHP adalah pemrograman interpreter yaitu penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. Menurut pendapat diatas dapat disimpulkan bahwa PHP adalah bahasa pemograman yang digunakan bersamaan dengan HTML dimana bahasa tersebut dijalankan di sisi *web browser*.

2.6 Pengertian MySQL

Menurut Arizona (2017) MySQL adalah bahasa yang digunakan untuk mengelola data pada RDBMS. Sedangkan menurut Risdiansyah (2017) MySQL merupakan *database server* yang bersifat *multiuser* dan *multi-threaded*.SQL adalah bahasa *database* standar yang memudahkan penyimpanan, pengubahan dan akses informasi. Pada MySQL dikenal istilah *database* dan tabel. Tabel adalah sebuah struktur data dua dimensi yang terdiri dari baris-baris *record* dan kolom.

2.7 Pengertian Basis Data

Basis data adalah kumpulan informasi yang disimpan didalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Pengertian basis data menurut Rosa dan Shalahuddin (2018), sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.

2.8 Pengertian HTML

Menurut Ardhana dalam Kesuma dan Rahmawati (2017) mengatakan bahwa HTML merupakan suatu bahasa yang dikenali oleh *web browser* untuk menampilkan informasi seperti teks, gambar, suara, animasi bahkan video.

2.9 Pengertian *Cloud Server*

Menurut Christina (2016), *cloud server* adalah sebuah *server virtual* yang berjalan di lingkungan *cloud computing. Cloud server* membangun, menyimpan, dan mengirim setiap data via platform *cloud computing* melalui internet. *Cloud server* memiliki seluruh *software* yang dibutuhkan untuk menjalankan dan bisa berfungsi sebagai unit yang independen.

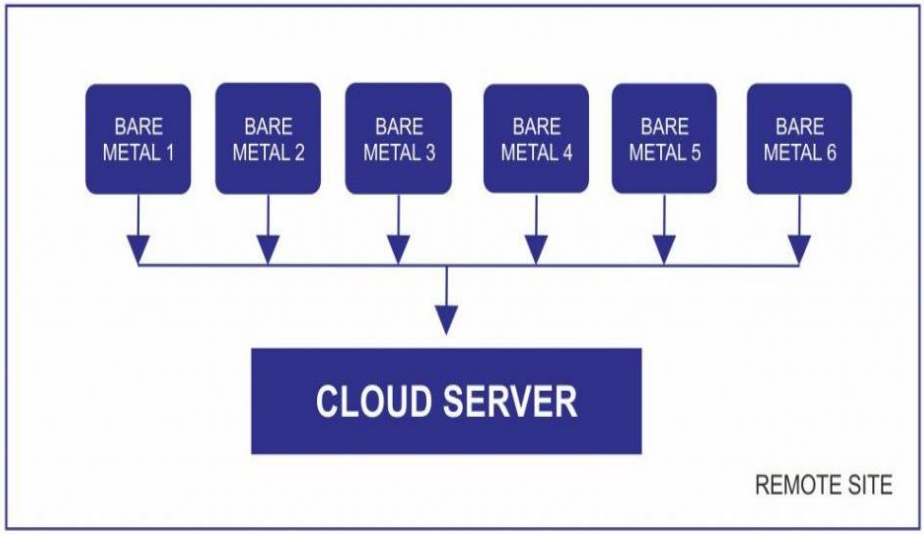
Sementara itu, *cloud* sendiri merupakan istilah yang sering digunakan untuk beberapa *server* yang tersambung dengan internet yang disewakan sebagai sebuah *software* atau aplikasi. Layanan berbasis *cloud* bisa termasuk *web hosting*, *data hosting*, atau *software*. *Cloud* juga bisa merujuk pada *cloud computing*, di mana beberapa *server* tersambung menjadi satu kesatuan untuk membagikan data. Itu artinya ketimbang menggunakan satu *server*, sebuah proses yang kompleks bisa didistribusi via komputer-komputer yang lebih kecil.

2.9.1 Kelebihan *Cloud Server*

Sebuah *cloud server* dapat memberikan perusahaan stabilitas dan keamanan karena masalah *software* yang bisa saja terjadi terisolasi dari lingkungan Anda. *Server cloud* lain tidak akan berdampak pada *server* Anda atau

sebaliknya. Jika seorang pengguna lainnya terlalu membebani server, ini tidak akan berdampak pada *cloud server* Anda tidak seperti *server* fisik.

- *Cloud server* stabil, cepat, dan aman. *Cloud server* bisa menghindari isu pada *hardware* yang biasa terjadi pada *server* fisik. *Cloud server* juga opsi paling stabil untuk perusahaan yang menginginkan *budget* mereka tetap minimal.
- *Cloud server* juga sangat menguntungkan. Anda akan mendapatkan lebih banyak sumber daya dan layanan lebih cepat daripada Anda mengeluarkan jumlah dana yang sama untuk sebuah *server* fisik. Website yang *cloud-hosted* juga akan bisa berjalan lebih cepat.
- Anda akan mendapatkan skalabilitas dengan penggunaan *cloud server*. *Cloud server* sangat cepat dan mudah *di-upgrade* dengan menambahkan memori dan ruang penyimpanan. Selain itu, *cloud server* juga terbukti lebih terjangkau.



Gambar 2.1 *Cloud Server*

2.9.2 Cara Kerja *Cloud Server*

Cloud server adalah *server* virtual yang *di-deployed* in lingkungan *cloud*. Ini bisa dilakukan dengan mengombinasi sebuah kumpulan *server* menjadi satu lalu mendistribusikan sumber daya ke komputer individu.

Cloud server bekerja dengan menggabungkan beberapa *server* menjadi satu *server* raksasa. Contohnya, jika ada 20 *server* di pusat data, mereka akan dikombinasi seolah adalah satu *server* raksasa. Anda bisa bayangkan layaknya mengombinasi beberapa *flashdrive*. Sekarang karena sudah ada sumber daya komputasi dalam jumlah besar, ruang penyimpanan dapat dengan mudah didistribusi ke orang-orang.

2.9.3 Perbedaan *Dedicated* vs *Cloud Server*

Di masa sekarang, sebuah perusahaan memiliki dua opsi apakah memilih penggunaan *cloud server* atau *dedicated server*. Berikut perbedaan mencolok antara dua hal tersebut:

2.9.3.1 Biaya operasional

Cloud server*: Cloud server* memiliki keunggulan berupa biaya yang lebih terjangkau dan tidak perlu pembayaran di awal untuk membeli atau *maintenance* infrastruktur. *Cloud* bisa dikonfigurasi untuk memberikan bisnis Anda fitur yang sama dengan *dedicated server. Cloud* lebih bisa diandalkan karena berjalan di beberapa *server*. Bahkan jika ada salah satu komponen yang *fail*, pelayanan tetap bisa berjalan dari *server* lainnya. *Cloud* tersedia di internet dan bisa membuat penggunanya mengakses data dari lokasi mana pun. Pengguna membayar berdasarkan layanan yang mereka gunakan.

Dedicated server: *Dedicated server* bukan hanya memiliki biaya *setup* yang mahal, tapi juga membutuhkan ongkos tambahan untuk *maintenance*. Sebuah perusahaan yang irit secara finansial harus menghindari *dedicated server* karena biaya di awal yang sangat mahal. Lebih lanjut, *server-server* ini membutuhkan administratornya masing-masing. Kurangnya jumlah administrator akan menjadi mimpi buruk karena harus mengonfigurasi lebih dari satu *server*.

2.9.3.2 Manajemen

***Cloud server*:** Meskipun *dedicated server* juga menawarkan kemudahan, tapi tetap tidak semudah mengatur sebuah *cloud server*. Manajemen *cloud server* sudah menjadi porsi kerjaan vendor.

Dedicated server: *Dedicated server* adalah sebuah *server* fisik dan Anda memiliki semua *server* untuk website Anda sendiri. *Dedicated server* jelas lebih aman dan memiliki performa yang baik karena Anda memiliki kendali penuh dan karyawan IT yang berdedikasi sepenuhnya untuk mengontrol *server*.

2.9.3.3 Reliability

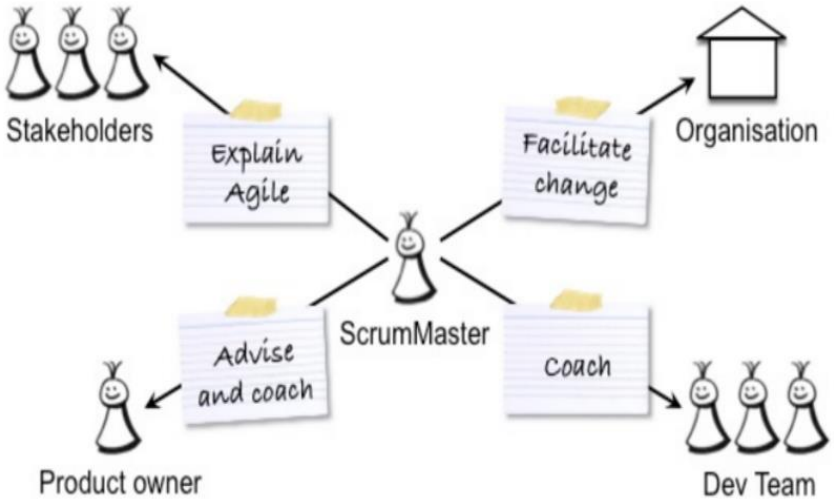
Cloud server*: Cloud server* mampu membuat beberapa instansi data. Jika dibutuhkan, data dari *server bare metal* bisa digunakan untuk memenuhi kebutuhan klien, sehingga bisa memastikan kelanjutan pengiriman konten ke *database* klien.

Dedicated server*: Dedicated server* seperti memiliki pulau sendiri, sehingga lebih aman. Akan tetapi, *dedicated server* memiliki satu kelemahan. Satu kesahalan kecil bisa membuat seluruh sistem *down* atau lebih buruknya mati total. Dengan *dedicated server*, Anda perlu *backup* database untuk mem-*backup server* lalu bisa memulihkannya jika ada bencana terjadi.

2.10 Metode *Scrum*

Metode atau model *scrum* merupakan metode turunan dari metodologi Agile. Metode *scrum* sudah dikembangkan oleh Jeff Sutherland sejak tahun 1993 yang tujuannya adalah menjadikan *scrum* sebagai metode pengembangan dan manajemen metodologi yang mengikuti prinsip-prinsip metodologi Agile T. Wijaya (2018). Metode *scrum* merupakan metode pengembangan perangkat lunak yang responsive dan berbasis metodologi agile yang mampu memberikan *value* dan manfaat yang terbaik terhadap suatu perangkat lunak yang akan dikembangkan. Metode *scrum* lebih berfokus pada manajemen proyek yang dimana keadaannya dalam perencanaan awalnya menyusahkan, komponennya belum begitu jelas sehingga menggunakan *feedback* berulang dalam mengontrol setiap fase atau prosesnya. Didalam *scrum* memiliki 3 *role* utama yaitu: *product*

owner, *scrum master*, dan *development team* atau tim pengembang. Yang dimaksud *Product Owner* (PO) atau pemilik produk, merupakan seorang profesional yang memiliki pengetahuan yang mendalam mengenai produk yang akan dikembangkan. Sementara itu *scrum team* atau tim development (Tim Pengembang), yang beranggotakan indvidu-individu yang mampu mengorganisasikan dirinya sendiri dan merupakan sebuah tim multifungsi dalam pengembangan suatu perangkat lunak. Kemudian yang dimaksud *scrum Master*, adalah orang yang bertanggung jawab untuk mengarahkan tim *scrum* (*scrum* team) bekerja menuju arah dan tujuan yang benar dan mengatasi semua rintangan dan halangan yang dapat mempengaruhi kinerja mereka dalam membuat suatu perangkat lunak.



Gambar 2.2 *Rule* pada Metode *Scrum*

Ketiga tahapan ini bekerja dalam proses pengembangan yang terus-menerus didalam sebuah iterasi yang disebut Sprint. Metode *scrum* memiliki beberapa langkah dalam pelaksanaannya yaitu:

1. Pengimplementasian *Product Backlog* yang berisi daftar tasks yang harus dikerjakan dalam proyek tersebut, perkiraan waktu yang dibutuhkan, dan jumlah tim yang terlibat.
2. Pengimplementasian perhitungan rencana *Design Sprint*.
3. Hasil perhitungan dari *Design Sprint*.
4. Analisis proses metode *scrum* dalam bentuk *Burndown Chart*.

Kemudian pada aktifitasnya *scrum* memiliki pendekatan yang hampir sama dengan langkah-langkah dalam pelaksanaan *scrum*. Pendekatan *scrum* memiliki terdiri dari aktifitas, yaitu:

1. *Backlog*.

Backlog adalah list dari fitur – fitur atau kebutuhan yang diinginkan oleh client.Pada backlog ini bisa bertambah atau berkurang.

2. *Sprints*.

Sprints merupakan iterasi atau fase kerja yang diperlukan untuk menyelesaikan list pada backlog sesuai dengan waktu yang ada pada *time-box*.

3. *Scrum* Meetings

Aktifitas *scrum* meetings merupakan meeting rutin atau pertemuan yang rutin dilakukan setiap hari pada pagi hari untuk mengevaluasi apa yang sedang dikerjakan, kendala yang ada, dan target penyelesaian *task* selanjutnya.

4. *Demo*

Memberikan peningkatan perangkat lunak kepada klien yang telah diimplementasikan dapat ditunjukan dan dievaluasi oleh klien.

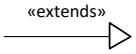
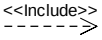
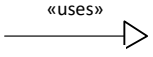
2.11 *Tools* / Alat Analisis Sistem

Alat Bantu perancangan sistem yang dapat digunakan adalah :

2.1.1.1 *Use Case Diagram*

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. Syarat penamaan pada *Use Case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami (Rosa dan Shalahuddin, 2016).

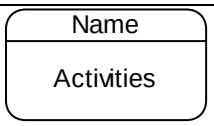
Tabel 2.1 Tabel *Use Case Diagram*

Keterangan	Simbol	Deskripsi
Use Case		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; dinyatakan dengan menggunakan kata kerja diawal-awal frase nama *Use Case*
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
Asosiasi		Komunikasi antara aktor dan *Use Case* memiliki interaksi dengan aktor
Ekstensi		Relasi *Use Case* tambahan ke sebuah *Use Case*, dimana *Use Case* yang ditambahkan dapat berdiri sendiri walau tanpa *Use Case* tambahan itu; mirip dengan prinsip *inheritance* pada pemograman berorientasi objek.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *Use Case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Include/uses	 include  Uses	Ada dua sudut pandang yang cukup besar mengenai include di *Use Case* : a. Include berarti *Use Case* yang ditambahkan akan selalu dipanggil saat *Use Case* tambahan dijalankan b. Include berarti *Use Case* yang tambahan akan selalu melakukan pengecekan apakah *Use Case*yang ditambahkan telah dijalankan sebelum *Use Case* tambahan dijalankan.

2.1.1.2 *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem (Rosa dan Shalahuddin, 2016). Simbol-simbol yang terdapat pada activity diagram dapat dilihat pada table dibawah ini.

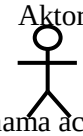
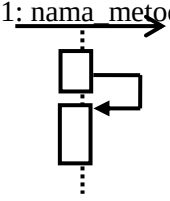
Tabel 2.2 Tabel *Activity Diagram*

Keterangan	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		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.1.1.3 *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat *Sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *Use Case* (Salahuddin, 2016). Adapun *symbol sequence diagram* dapat diliat pada table dibawah ini.

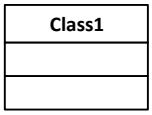
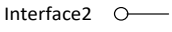
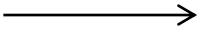
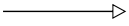
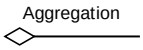
Tabel 2.3 Tabel *Sequence Diagram*

Simbol	Deskripsi
Aktor  nama actor Atau nama aktor Tanpa waktu aktif	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
Garishidup/ lifeline	Menyatakan kehidupan suatu objek
Objek nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
Pesan tipe create <<create>>	Menyatakan suatu objek membuat objek lain, arah panah objek yang dibuat
Pesan tipe call 1: nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri 1 : nama_metode()
Pesan tipe send 1 : Masukkan	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi
Pesan tipe return 1 : keluaran	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
Pesan tipe destroy <<destroy>>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy

2.1.1.4 *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. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa dan Shalahuddin, 2016). Simbol-simbol yang ada pada *class diagram* dapat dilihat dibawah ini.

Tabel 2.4 Tabel *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
atarmuka/interface 	Sama dengan konsep interface dalam pemograman berorientasi objek.
Asosiasi	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan multiplicity.
Asosiasi berarah 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi 	Relasi antar kelas dengan makna semua bagian (whole-part).