

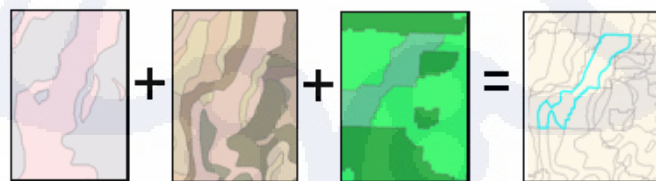
BAB II DASAR TEORI

2.1 *Sistem Informasi Geografis (SIG)*

2.1.1 Konsep Dasar

Sistem informasi adalah rangkaian proses yang dilakukan pada suatu data untuk menghasilkan informasi yang dapat dimanfaatkan untuk pengambilan keputusan. Sistem Informasi Geografis (SIG) merupakan bentuk sistem informasi yang diterapkan pada bentuk data geografis, yakni data yang mengandung informasi spasial yang berkaitan dengan ruangan (*space*) yang ditempati suatu objek dimana objek tersebut memiliki keterhubungan antar elemen dalam suatu sistem. Definisi sistem informasi geografis secara umum yaitu suatu sistem komputer yang mampu menangkap, menyimpan, menganalisis, dan menampilkan informasi, baik informasi yang direferensi secara geografis maupun informasi non-spasial, yang dapat digunakan untuk mendukung pengambilan keputusan. [GEO97]

Kekuatan dalam SIG yaitu adanya teknik pengolahan data yang dapat mengkombinasikan dua atau lebih data spasial yang berbeda untuk menghasilkan data atau informasi baru. Teknik seperti ini disebut *overlay*. Dengan *overlay*, data dengan konteks spasial berbeda dapat dihubungkan dan diintegrasikan menjadi informasi baru untuk membantu proses pengambilan keputusan. *Overlay* di sini menyerupai konsep *overlay* yang diterapkan pada diagram venn dalam konteks matematika, yang berkaitan dengan operasi yang dilakukan terhadap dua objek yang saling bertumpuk seperti operasi *union*, interseksi, *clip*, dan lain sebagainya. Bentuk *overlay* untuk mengintegrasikan data spasial diperlihatkan pada Gambar II-1.



Gambar II-1 Teknik *Overlay* pada *Dataset*

2.1.2 Komponen SIG [MAI04]

Pada SIG, terdapat lima komponen yang harus bekerja secara sinergis agar sistem dapat berjalan dengan baik. Komponen tersebut yaitu *Hardware*, *Software*, *Data*, *People*, dan *Methods/Procedures* (Gambar II-2).



Gambar II-2 Komponen SIG

Berikut deskripsi untuk tiap komponen SIG:

1. *Hardware*

Hardware terdiri atas perangkat teknis yang dibutuhkan untuk menjalankan SIG. Perangkat tersebut meliputi sistem komputer dengan *power* yang cukup untuk menjalankan perangkat lunak, mempunyai memori yang cukup untuk menyimpan data dalam jumlah besar, dan peralatan *input-output* seperti *scanner*, *digitizer*, dan *printer*.

2. *Software*

Software SIG yaitu perangkat lunak SIG yang mampu menyimpan dan mengolah data grafis dan data deskriptif untuk disimpan pada basis data tunggal. Sebelum ada inovasi ini, model geo-relasional digunakan, di mana data grafis dan data deskriptif ditangani secara terpisah.

3. *Data*

Data merupakan salah satu faktor penting dalam SIG, karena pada dasarnya SIG bertugas untuk mengolah data supaya tercipta informasi yang bermanfaat bagi penggunanya. Data yang diolah pada SIG meliputi data raster, data vektor, dan data non-spasial yang mengacu terhadap kedua data sebelumnya. Penjelasan lebih rinci mengenai data pada SIG diberikan pada subbab 2.1.3.

4. *People*

Yang dimaksud '*people*' dalam konteks ini yaitu orang-orang yang berinteraksi secara langsung dengan SIG. Orang-orang yang berinteraksi dengan SIG dapat dikategorikan menjadi tiga, yaitu *viewers*, *general users*, dan ahli SIG.

Viewers adalah masyarakat luas yang hanya menggunakan SIG untuk melihat-lihat (*browse*) data geografis sebagai bahan referensi. *General Users* adalah orang yang menggunakan SIG untuk membangun bisnis, memberikan layanan profesional, dan membuat keputusan. Pengguna kelas ini meliputi manajer, perancang, ilmuwan, teknisi, pengusaha, dan

sebagainya. Ahli SIG adalah orang-orang yang terlatih yang dapat membuat sistem ini bekerja. Orang tersebut meliputi manajer SIG, *database administrators*, *application specialists*, *systems analysts*, dan *programmers*. Ahli SIG bertanggung jawab dalam pemeliharaan data geografis dan penyediaan dukungan teknis SIG.

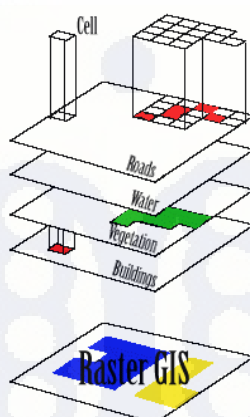
5. *Methods* / Prosedur

Prosedur adalah langkah-langkah yang diambil untuk menjawab pertanyaan yang memerlukan penyelesaian. Prosedur meliputi bagaimana data akan di-*retrieve*, dimasukkan ke dalam sistem, disimpan, dikelola, diubah, dianalisis, dan ditampilkan.

2.1.3 Representasi Data SIG

SIG merepresentasikan objek-objek pada dunia nyata seperti jalan, penggunaan lahan, atau kemiringan lahan dalam bentuk data digital. Objek-objek pada dunia nyata tersebut dapat digolongkan menjadi dua, yaitu objek diskrit dan objek kontinu. Objek diskrit yaitu objek yang terdiri dari sejumlah hingga berhingga elemen yang berbeda. Objek diskrit mudah diidentifikasi elemen-elemen pembentuknya karena mempunyai batas yang jelas, beberapa contoh objek diskrit diantaranya yaitu rumah, jalan, dan sungai. Objek kontinu yaitu objek yang terdiri dari nilai yang tak berhingga dan berkesinambungan. Yang termasuk dalam objek kontinu yaitu kemiringan lahan dan curah hujan. Untuk merepresentasikan objek dari dunia nyata menjadi data SIG dalam bentuk digital, terdapat dua metode penyimpanan data yang digunakan, yaitu raster dan vektor.

1. Data Raster

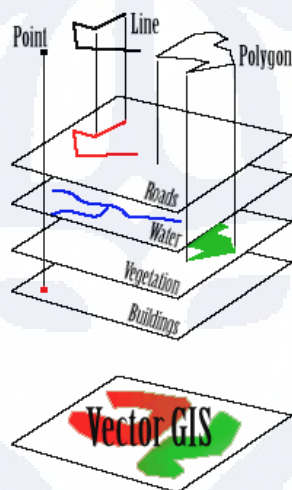


gambar II-3 Data Raster pada SIG

Data raster adalah data yang dibentuk oleh matriks yang terdiri dari sel/piksel yang diorganisasikan dalam baris dan kolom, di mana tiap selnya mengandung nilai yang merepresentasikan informasi. Data yang disimpan dalam format raster dapat berupa data diskrit, kontinu, gambar hasil *scanning* atau foto. Resolusi data raster ditentukan oleh lebar sel

pembentuknya. Data raster disimpan dalam berbagai format antara lain format TIF, BMP, dan JPEG. [GPL99]

2. Data Vektor



gambar II-4 Data Vektor pada SIG

Tipe data vektor yaitu data yang direpresentasikan ke dalam tiga bentuk geometri vektor, yaitu titik (*point*), garis (*line*), dan area (*polygon*). Fitur geografis yang dapat direpresentasikan dalam bentuk titik antara lain gunung, menara, serta titik kemiringan, dalam bentuk garis antara lain sungai dan jalan, serta dalam bentuk area yaitu bentuk-bentuk rupa bumi seperti pulau, batas kota, atau bangunan. Setiap fitur yang direpresentasikan oleh salah satu bentuk geometri tersebut dihubungkan ke baris-baris pada database yang mendeskripsikan atribut (tabel atribut) setiap fitur. [GPL99]

Tipe data vektor dan raster memiliki kelebihan dan kekurangan masing-masing. Himpunan data raster mencatat nilai untuk setiap titik pada data tersebut, sehingga memerlukan ruang penyimpanan yang lebih besar dibandingkan representasi data pada format vektor. Ukuran data raster bisa mencapai 10 hingga 100 kali lebih besar dibanding ukuran data vektor, bergantung pada tingkat resolusinya. Data vektor lebih kompatibel dengan lingkungan basis data relasional, dan mudah di-*update*. Meski demikian, lebih mudah mengimplementasikan *overlay* dengan menggunakan data raster dibandingkan dengan menggunakan data vektor.

2.2 SIG Mobile

2.2.1 Konsep Dasar

Komputasi *mobile* membuat suatu perubahan mendasar dengan menambahkan kemampuan untuk dapat menggunakan teknologi SIG langsung di lapangan sehingga informasi geografis

dapat diakses kapan pun dan di mana pun dengan informasi yang lebih akurat dan selalu *ter-update*. Hal ini memungkinkan manusia dapat melakukan analisis terhadap informasi geografis melalui interaksi secara langsung dengan lingkungan sekitarnya. Teknologi tersebut dikenal dengan teknologi SIG *mobile*, yaitu teknologi yang mengaplikasikan penggunaan SIG pada perangkat mobile seperti laptop, PDA, dan telepon selular [PEN03]. SIG mobile dapat merupakan integrasi dari satu atau lebih teknologi-teknologi berikut:

1. Sistem Informasi Geografis
2. Perangkat *mobile*
3. *Global Positioning Systems* (GPS)
4. Komunikasi nirkabel untuk pengaksesan SIG melalui internet

[ESR07]

Meskipun demikian, dibalik dari kelebihan-kelebihannya, SIG *mobile* tetap memiliki keterbatasan terkait dengan *mobile device* dan koneksi *wireless*-nya. *Mobile device* dengan ukuran yang relatif kecil menyebabkan keterbatasan akan konsumsi daya (*power*), memori, dan layar antarmuka. Kekurangan lainnya yaitu untuk arsitektur SIG *mobile* yang menggunakan koneksi nirkabel. Koneksi nirkabel memungkinkan kemungkinan kehilangan konektivitas tinggi akibat banyaknya gangguan di udara, pembatasan frekuensi oleh regulasi yang ada, bandwidth yang relatif kecil, *delay* tinggi, dan keamanannya rendah karena menggunakan *shared medium*.

2.2.2 Global Positioning System [GAR08]

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang dibuat oleh 24 jaringan satelit yang ditempatkan pada orbit oleh U.S. *Department of Defense*. GPS pada mulanya dibuat untuk aplikasi militer, tetapi pada tahun 1980 pemerintah memperbolehkan penggunaan sistem tersebut untuk masyarakat sipil. GPS dapat bekerja pada berbagai cuaca, di mana pun, dan selama 24 jam sehari, serta tidak dikenakan biaya.

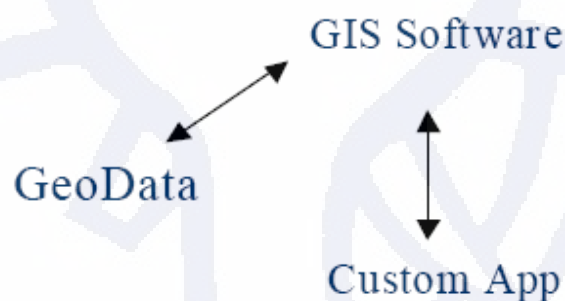
Satelit GPS mengelilingi bumi dua kali sehari melalui orbitnya untuk mengirimkan sinyal informasi ke bumi. GPS *receiver* menerima informasi ini dan menggunakan *triangulation* untuk menghitung posisi penggunanya. Pada dasarnya, GPS *receiver* membandingkan antara waktu pengiriman sinyal dari satelit dengan waktu sinyal diterima oleh *receiver*. Perbedaan waktu ini akan memberitahukan GPS *receiver* seberapa jauh jarak satelit. Dengan melakukan pengukuran jarak dari beberapa satelit, GPS *receiver* dapat menentukan posisi *user* dan menampilkannya pada peta *digital*.

GPS *receiver* harus mengunci sinyal setidaknya dari tiga satelit untuk dapat menghitung posisi dan melacak pergerakan *user* dalam 2 dimensi (*latitude* dan *longitude*). Dengan menggunakan empat atau lebih satelit, *receiver* akan mampu menentukan posisi *user* dalam 3 dimensi (*latitude*, *longitude*, dan *altitude*). Setelah posisi *user* ditentukan, unit GPS dapat menghitung informasi lainnya, seperti kecepatan, muatan, jejak (*track*), jarak perjalanan, jarak ke tempat tujuan, waktu matahari terbit/terbenam, dan sebagainya.

2.2.3 Arsitektur SIG Mobile

Arsitektur SIG *mobile* dapat diklasifikasikan menjadi lima, yaitu *Stand-Alone Client*, *Client-Server*, *Distributed Client-Server*, *Services*, dan *Peer to Peer* [HAS04].

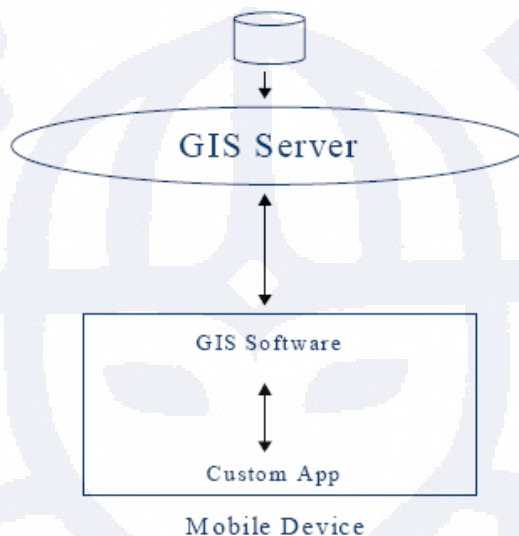
1. *Stand-Alone Client*



Gambar II-5 Arsitektur *Stand-Alone Client*

Stand-Alone Client merupakan arsitektur SIG *mobile* yang paling sederhana. Pada arsitektur ini, seluruh aplikasi berada pada *mobile device*. *Device* menyimpan data geografis (*geodata*), *software* SIG *mobile* untuk membaca dan menampilkan data, serta *custom application* yang dibangun di atas *software* SIG seperti tampak pada Gambar II-5. Pendekatan lainnya yaitu dengan mengembangkan *custom application* yang mampu membaca dan menampilkan *geodata* secara langsung, tanpa memerlukan *software* SIG secara terpisah. Arsitektur jenis ini memiliki keterbatasan. Pertama, *resource hardware* dari *mobile device* membatasi jumlah *geodata* yang dapat di-*support* oleh aplikasi. Kedua, arsitektur ini tidak memungkinkan komunikasi dengan aplikasi lain ataupun kolaborasi dengan aplikasi sejenis.

2. *Client-Server*

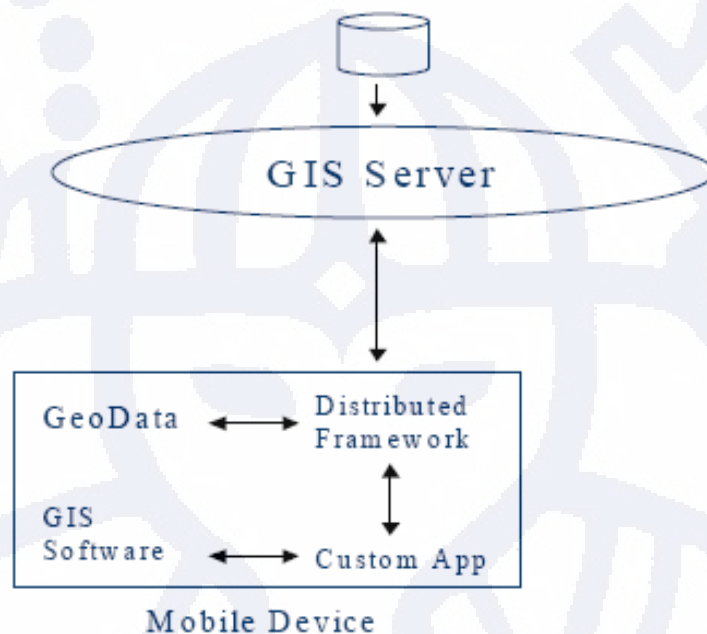


Gambar II-6 Arsitektur *Client-Server*

Arsitektur *client-server* merupakan salah satu bentuk arsitektur yang dapat menutupi keterbatasan yang ada pada arsitektur *stand-alone client*. Pada arsitektur ini *geodata* disimpan dalam komputer yang terpisah dari *mobile device* dan disajikan kepada *client* melalui *server* SIG. *Customized application* sama dengan arsitektur *stand-alone client*, hanya perlu diubah pada konfigurasi *software* SIG-nya, di mana sekarang *software* SIG harus mengambil *geodata* dari *server* SIG.

Kelebihan arsitektur ini dari arsitektur *stand-alone client* yang pertama yaitu jumlah *geodata* dibatasi oleh *resource server* yang secara virtual tidak terbatas. Kedua, *multiple mobile device* yang menjalankan aplikasi yang sama dapat mengakses *server* secara bersamaan, yang memungkinkan arsitektur *multi-user* yang potensial. Kekurangan dari arsitektur ini yaitu perlunya konektivitas yang baik. Apabila konektivitas tidak mendukung dikarenakan faktor jarak, interferensi, serta faktor penghalang lainnya, *mobile device* tidak akan dapat berkomunikasi dengan *server* SIG sehingga aplikasi tidak dapat mengakses *geodata*. Karena konektivitas seringkali tidak konsisten dan menjadi permasalahan umum pada aplikasi *mobile*, penggunaan arsitektur ini juga perlu dipertimbangkan.

3. *Distributed Client Server*



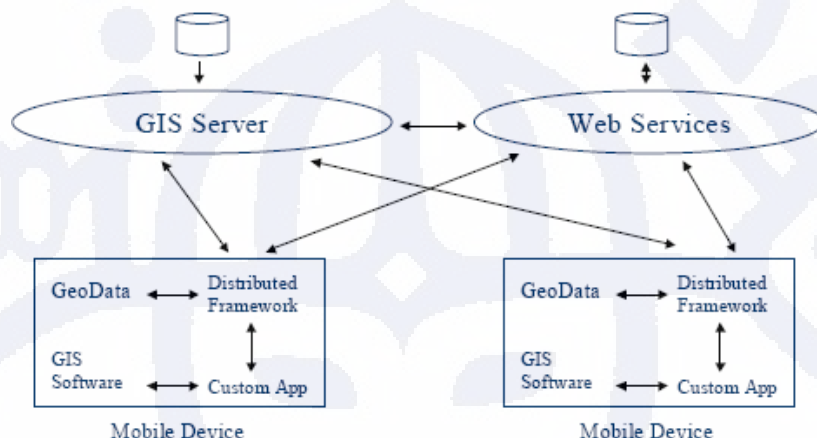
Gambar II-7 *Arsitektur Distributed Client Server*

Untuk mengatasi permasalahan ketidakkonsistenan konektivitas pada arsitektur *client server*, diperlukan dua konsep sistem terdistribusi, yaitu *persistence* dan *resource management*.

1. *Persistence* : ketika *mobile device* tidak dapat terhubung ke *server*, secara persisten *mobile device* mencoba untuk *reconnect*.
2. *Resource Management* : ketika *mobile device* tidak dapat terhubung ke *server*, *mobile device* dapat menggunakan sekumpulan *cached data* yang ditempatkan pada *mobile device*.

Komponen yang dapat menangani *persistence* dan *resource management* disebut *distributed framework*. Arsitektur ini dapat mendukung sebagian aplikasi SIG mobile agar *robust* dan *reliable*, tetapi, tidak memungkinkan *extensibility* pada *back end*.

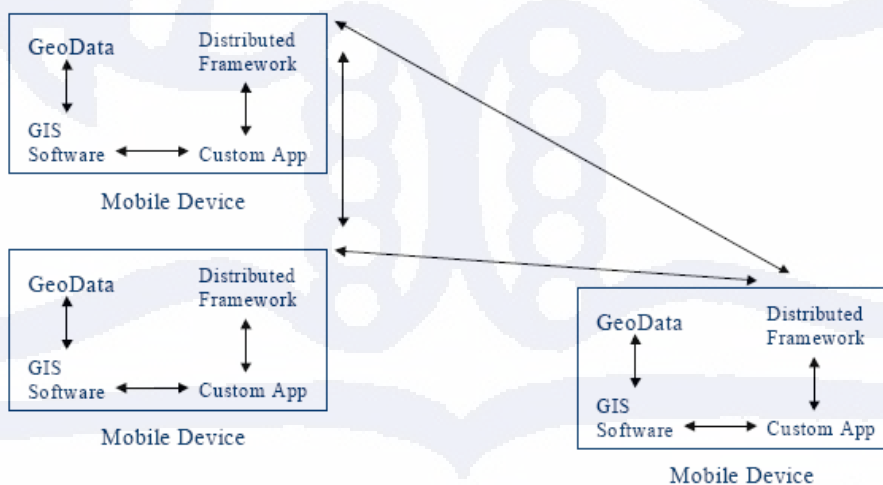
4. Services



Gambar II-8 Arsitektur Services

Untuk dapat menyediakan kemampuan ekstensibilitas *back-end*, arsitektur *services* memandang *server* SIG sebagai *web service* dan memungkinkan *web service* lainnya menjadi bagian dari aplikasi. Selama *web service* menggunakan protokol komunikasi yang sama, *mobile device* dapat berkomunikasi dengan mereka. *Web service* juga dapat berkomunikasi dengan *web service* itu sendiri. Protokol komunikasi umum yang digunakan adalah SOAP XML, yaitu standar yang digunakan untuk bertukar pesan antar komponen *software*. Arsitektur ini mendukung komunikasi yang *robust* antara sejumlah *mobile device* dan juga *web service*, tetapi hal ini mungkin tidak ideal untuk beberapa aplikasi seperti halnya aplikasi yang dirancang untuk kolaborasi dengan *remote area* di mana tidak tersedia konektivitas ke *server*.

5. Peer to Peer



Gambar II-9 Arsitektur Peer to Peer

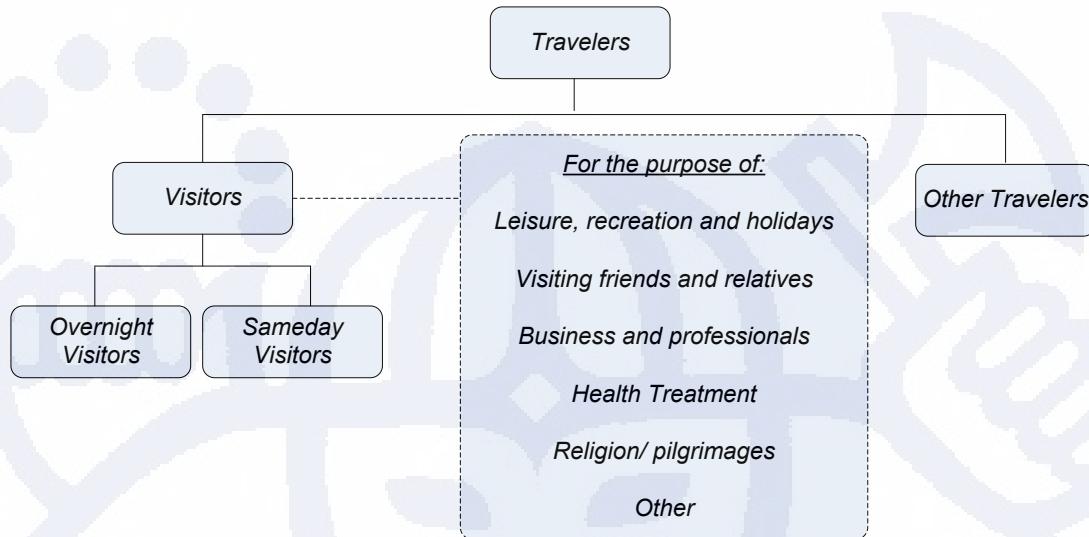
Arsitektur *peer-to-peer* juga merupakan arsitektur alternatif yang dapat dipilih untuk mengatasi keterbatasan yang ada pada arsitektur *stand-alone client*, karena arsitektur ini memungkinkan komunikasi antara *mobile device*. Karena *server* tidak tersedia untuk menyimpan *geodata*, data harus disimpan di dalam *device*. Tetapi jika setiap *mobile device* menyimpan 100% data, arsitektur ini akan memiliki keterbatasan yang sama dengan arsitektur *stand-alone client*. Untuk memungkinkan lebih banyak tempat penyimpanan data, tiap *mobile device* hanya menyimpan beberapa bagian *geodata*.

Sebagai ilustrasi, ketika *mobile device* A memerlukan data, *mobile device* dapat mencari data tersebut melalui *resource management* dari *distributed framework* untuk mengetahui apakah data yang diperlukan ada pada tempat penyimpanan lokal. Jika tidak tersedia, *framework* tersebut harus mengetahui bagaimana caranya untuk mengakses data dari *mobile device* B. Oleh karena itu, arsitektur ini perlu penggunaan konsep sistem terdistribusi lainnya, yaitu *naming-unique identifier* untuk tiap *device*, untuk dapat membedakan antar *device*. Akan tetapi, apabila *device* B berada di luar jangkauan A, maka A tidak dapat mengakses data yang diperlukan. Untuk mengatasi permasalahan ini, arsitektur ini perlu menggunakan konsep sistem terdistribusi yang terakhir, yaitu *redundancy*, yang dapat *maintain* data pada satu atau lebih *device* sebagai alternatif penyimpanan data, sehingga, *distributed framework* pada *device* A dapat mengetahui bahwa data yang dicari dapat juga diperoleh dari *mobile device* C.

2.3 Turisme

2.3.1 Turis

Turisme sangat berperan meningkatkan ekonomi suatu daerah. Oleh karena itu, penelitian telah banyak dilakukan untuk menganalisis segala hal terkait dengan turis. Untuk keperluan tersebut, para ahli juga telah melakukan penelitian untuk mendefinisikan konsep turisme, salah satunya adalah konsep turisme yang didefinisikan oleh *World Tourism Organisation* (1995). WTO mengembangkan skema klasifikasi *travelers* seperti tampak pada Gambar II-10.



Gambar II-10 Visitors dan Other Travelers

Pada klasifikasi *travelers* yang dikembangkan WTO, turis merupakan salah satu kategori dari *visitors*, yakni *overnight visitor*. Penjabaran lengkap mengenai turisme berdasarkan definisi dan klasifikasi oleh WTO dapat dilihat pada Tabel II-1.

Tabel II-1 Definisi dan Klasifikasi Turisme berdasarkan WTO [WTO95]

Tourism	Aktivitas dari orang yang melakukan perjalanan ke atau tinggal di tempat-tempat di luar lingkungannya selama tidak lebih dari satu tahun dengan tujuan untuk berekreasi, berbisnis, atau tujuan lainnya yang tidak terkait dengan aktivitas remunerasi
Tourist (Overnight visitor)	<i>Visitor</i> yang tinggal di tempat yang dikunjunginya selama paling sedikit satu malam dengan menggunakan akomodasi pribadi maupun kolektif.
Same day visitor (Excursionists)	<i>Visitor</i> yang menghabiskan waktu kurang dari satu malam di tempat yang dikunjunginya
Visitor	Orang yang melakukan perjalanan ke tempat lain di luar lingkungannya selama kurang dari satu tahun dan tidak mempunyai tujuan untuk bekerja di tempat yang dikunjunginya
Traveler	Orang yang melakukan perjalanan antara dua atau lebih lokasi yang berbeda

WTO juga mengemukakan bahwa terdapat dua elemen pada turisme: perjalanan turis dalam menuju tempat tujuan dan aktivitas turis ketika sampai di tempat tujuan. Perpindahan yang dilakukan turis menuju tempat tujuan sifatnya hanya sementara, dan turis akan berencana untuk kembali ke tempat asalnya dalam jangka waktu beberapa hari, minggu, atau bulan.

2.3.2 Pemandu Turis

Kegiatan turisme pada umumnya sangat berkaitan erat dengan tiga hal berikut: transportasi yang membawa turis dari satu tempat ke tempat lain, akomodasi untuk tempat tinggal turis, dan pelayanan yang menyediakan barang dan jasa untuk memenuhi kebutuhan para turis. Peranan pemandu turis menjadi penting bagi turis untuk membantu mereka dalam memperoleh transportasi, akomodasi, ataupun pelayanan yang sesuai dengan keinginan dan kebutuhan mereka. [UHE06]

European Federation of Tourist Guide Associations (FEG) mendefinisikan Pemandu Turis sebagai seseorang yang mempunyai kualifikasi pemandu turis untuk wilayah tertentu yang diakui oleh masyarakat yang berwenang di negara terkait. Peran utama Pemandu Turis adalah untuk mengiringi sekelompok atau individual turis baik yang berasal dari luar maupun dalam negeri untuk mengunjungi monumen, museum, atau tempat-tempat menarik lainnya yang terdapat pada kota atau wilayah tersebut, dengan mengartikan/memberi penjelasan kepada turis sesuai dengan bahasa yang dapat dipahami turis, mengenai warisan kebudayaan serta alam yang terdapat pada wilayah tersebut dengan cara menarik. [FEG99]

2.4 Rekayasa Interaksi

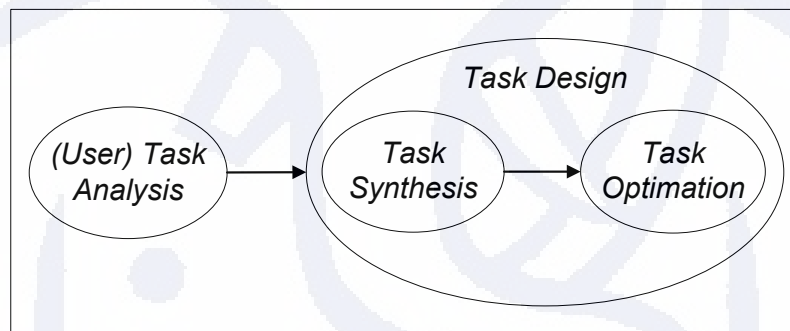
2.4.1 Konsep Dasar

Terdapat beberapa definisi mengenai perancangan interaksi (*interaction design*). [FOV07] mendefinisikan perancangan interaksi sebagai suatu proses yang sistematis dan iteratif untuk merancang antarmuka yang interaktif, dengan menggunakan metodologi meliputi penelitian dan penemuan teknik-teknik seperti analisis kebutuhan, analisis *stakeholder*, serta analisis *task*. Definisi perancangan interaksi berdasarkan [PRE02] adalah bagaimana merancang produk yang interaktif untuk mendukung manusia dalam kehidupannya sehari-hari. [FOR05] mendefinisikan perancangan interaksi sebagai perancangan mengenai bagaimana *user* berkomunikasi atau berinteraksi dengan komputer. Perancangan tersebut difokuskan pada jalannya interaksi, *dialog* antara manusia dengan komputer, hubungan antara masukan dan keluaran, serta mekanisme *feedback*.

Tujuan dari perancangan interaksi yaitu untuk menghasilkan produk interaktif yang mudah dipelajari, dapat digunakan secara efektif, dan *enjoyable* dari perspektif *user*. Untuk memperoleh produk yang interaktif tersebut, perancangan interaksi perlu menerapkan konsep *usability* dalam proses perancangannya [PRE02].

2.4.2 Pemodelan *Task*

Perancangan interaksi pada umumnya ditujukan untuk menggantikan atau memperbaiki sistem yang sudah ada ataupun untuk mengembangkan sistem inovatif benar-benar dibuat dari awal. Untuk membangun sebuah sistem yang baru, diperlukan adanya perancangan terhadap sistem baru agar diperoleh hasil yang sesuai dan dapat digunakan sebagaimana mestinya. Salah satu metode untuk memodelkan perancangan sistem yaitu pemodelan *task* [SAS06]. Proses pemodelan *task* pada perancangan interaksi sistem dapat dilihat pada Gambar II-11.



Gambar II-11 Tahapan Pemodelan *Tasks* pada Sistem HCI [SAS06]

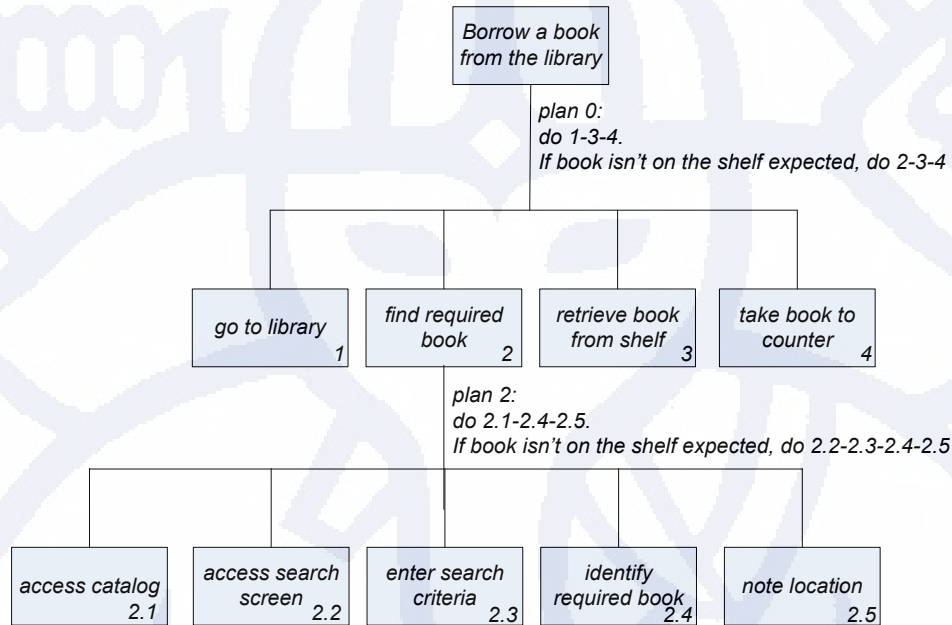
Berikut penjelasan rinci mengenai setiap tahapan pada pemodelan *task*:

1. *(User) Task Analysis*

Berdasarkan [PRE02], *task analysis* digunakan untuk memodelkan kondisi saat ini yang sudah ada, bukan kondisi yang diinginkan. *Task analysis* mencakup teknik untuk mencari pengertian proses serta aksi fisik pada tingkat abstraksi yang lebih tinggi dan lebih rinci. Terdapat beberapa teknik *task analysis*, diantaranya GOMS (*goals, operations, methods, and selection rules*), HTA (*hierarchical task analysis*), dan CTA (*cognitive task analysis*). Pada subbab ini, hanya akan dijelaskan mengenai teknik *task analysis* yang paling banyak digunakan, yakni HTA. [PRE02]

HTA dirancang untuk mengidentifikasi kebutuhan, dengan memecah *task* menjadi *sub task*, *sub-sub task*, dan seterusnya. *Task-task* tersebut kemudian disatukan menjadi suatu rencana yang menjelaskan bagaimana *task* dilakukan pada kondisi yang sebenarnya. HTA fokus pada aksi fisik yang dapat diamati, meliputi aksi yang tidak berkaitan dengan interaksi dengan perangkat lunak atau dengan *device* lainnya sekalipun (mengabaikan

teknologi yang digunakan). Pemecahan *task-task* tersebut didasari dari tujuan *user*. *Task-task* utama yang terbentuk merupakan langkah-langkah yang harus *user* lakukan untuk mencapai tujuannya. Selanjutnya, *task* utama tersebut dapat dipecah kembali menjadi *sub task* jika diperlukan. Contoh representasi grafis HTA dapat dilihat pada Gambar II-12.



Gambar II-12 HTA meminjam buku dari perpustakaan [PRE02]

Berdasarkan [SAS06], *task analysis* merupakan suatu proses analisis mengenai cara manusia (*user*) melakukan suatu pekerjaan (*tasks*). Berikut adalah teknik yang digunakan untuk membangun *task analysis*:

- Untuk setiap pengguna, lakukan analisis mengenai pekerjaan apa saja yang dilakukan terkait dengan persoalan dan untuk mencapai tujuan pekerjaan.
- Untuk setiap pekerjaan pengguna, lakukan dekomposisi (penguraian) pekerjaan ke dalam *tasks* dan *subtasks*; klasifikasi pengetahuan mengenai setiap *task* (aksi dan objek); dan organisasi atas pengetahuan tersebut.

Setiap model *task* direpresentasikan sebagai:

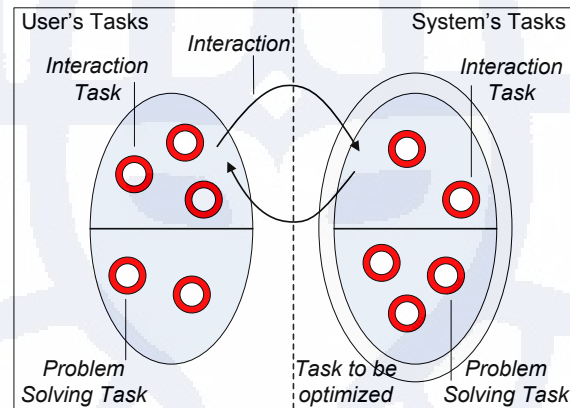
***task* <verb, noun>**,

verb menunjukkan aksi yang dilakukan *user* atau *system*

noun menunjukkan objek dari aksi

- Task design** merupakan perancangan untuk memperoleh model *tasks* yang representatif terhadap rancangan perangkat lunak (sistem) dan kontribusi pengguna. Proses ini terdiri dari dua tahap utama, yaitu: *task synthesis* (pemetaan, reduksi, dan alokasi terhadap *tasks*

pengguna dan sistem) dan *task optimization* (mengelola *task* sebagai komponen sistem ke dalam klasifikasi *generic*, *specific*, atau *parametric tasks* dengan harapan model sistem optimal). Pada tahap ini berarti telah mulai dilakukan perancangan *tasks* untuk sistem. *Tasks* yang sudah mulai dipetakan tersebut berorientasi pada penyelesaian persoalan (*problem solving tasks*) dan interaksi (*interaction tasks*) seperti yang ditunjukkan pada Gambar II-13.



Gambar II-13 Keterhubungan *User's Tasks* dan *System's Tasks* [SAS06]

2.4.3 Usability

Usability adalah kriteria yang dapat menilai seberapa mudah penggunaan antarmuka oleh *user* [NIE03]. *Usability* mengacu pada metode untuk meningkatkan kemudahan penggunaan dalam proses perancangan. *Usability* memiliki lima komponen kriteria sebagai berikut:

1. *Learnability*, yaitu seberapa mudah penggunaan sistem oleh *user* untuk menyelesaikan *task-task* dasar ketika pertama kali menggunakan.
2. *Efficiency*, yaitu seberapa cepat *user* dapat melakukan suatu *task* setelah pengguna mempelajari design-nya.
3. *Memorability*, yaitu ketika *user* kembali menggunakan desain tersebut setelah tidak menggunakannya dalam beberapa waktu, seberapa cepat *user* dapat mahir kembali menggunakan desain tersebut.
4. *Errors*, yaitu seberapa banyak kesalahan yang dilakukan *user*, seberapa jauh kesalahan tersebut, dan seberapa cepat *user* dapat pulih dari kesalahan tersebut.
5. *Satisfaction*, yaitu seberapa senang *user* menggunakan desain tersebut.

Terdapat empat paradigma untuk menggali kebutuhan *user* agar tujuan *usability* sistem dapat dicapai [PRE02], yaitu sebagai berikut:

1. *Quick and Dirty*

Pada paradigma ini, *designer* memperoleh *feedback* dari *user* secara informal untuk memperoleh kebutuhan *user*. Data yang diperoleh pada paradigma ini berupa data

deskriptif hasil *interview* langsung dari *user*. Cara ini digunakan untuk memperoleh *feedback user* dalam waktu singkat dan murah.

2. *Usability Testing*

Pada paradigma ini, kebutuhan *user* diperoleh dengan mengukur performansi *user* dengan meminta *user* melakukan sejumlah *task* yang telah dipersiapkan sebelumnya. Data yang diambil pada pengujian ini pada umumnya adalah jumlah kesalahan yang dilakukan *user* dan waktu yang diperlukan *user* dalam menyelesaikan setiap *task*.

3. *Field Studies*

Pada paradigma ini, kebutuhan *user* diperoleh dengan mengukur performansi *user* pada lingkungan yang sebenarnya (*natural*). Pengujian dengan paradigma ini biasanya dilakukan untuk menguji sistem yang sudah berjalan sebelumnya guna memperoleh kekurangan pada sistem lama untuk diperbaiki.

4. *Predictive Evaluation*

Pada paradigma ini, kebutuhan *user* diperoleh oleh para ahli dengan menerapkan pengetahuan yang dimilikinya mengenai *user* tertentu, sehingga diperoleh permasalahan *usability* yang mungkin dialami *user* tersebut tanpa perlu melibatkan *user* dalam perancangan sistem.

Dalam praktiknya, tingkat *usability* sistem perlu diukur untuk mengetahui seberapa jauh *usability* telah diterapkan pada suatu sistem. Menurut [PRE02], terdapat lima teknik pengujian *usability*, antara lain:

1. *Observing Users*. Dalam teknik ini, pengujian dilakukan dengan melakukan observasi pada *user* ketika menggunakan sistem.
2. *Asking Users*. Dalam teknik ini, pengujian dilakukan dengan menanyakan langsung kepada *user* mengenai kebutuhan dan permasalahan yang dialami *user* ketika menggunakan sistem.
3. *Asking Experts*. Dalam teknik ini, pengujian dilakukan dengan meminta para ahli (*expert*) untuk menilai efektivitas *usability* sistem.
4. *User Testing*. Dalam teknik ini, pengujian dilakukan dengan mengukur hasil perancangan kepada sejumlah *user* dengan meminta *user* melakukan sejumlah *task* tertentu pada lingkungan yang telah diatur sedemikian rupa menyerupai laboratorium.
5. *Modelling User's Task Performance*. Dalam teknik ini, pengujian dilakukan dengan memodelkan interaksi manusia dengan komputer untuk memprediksi efisiensi dan permasalahan terhadap hasil perancangan tanpa harus membangun prototipe perangkat lunak.