

# PENGANTAR TEKNIK KOMPILASI

- Tujuan

- Konsep Teknik Kompilasi

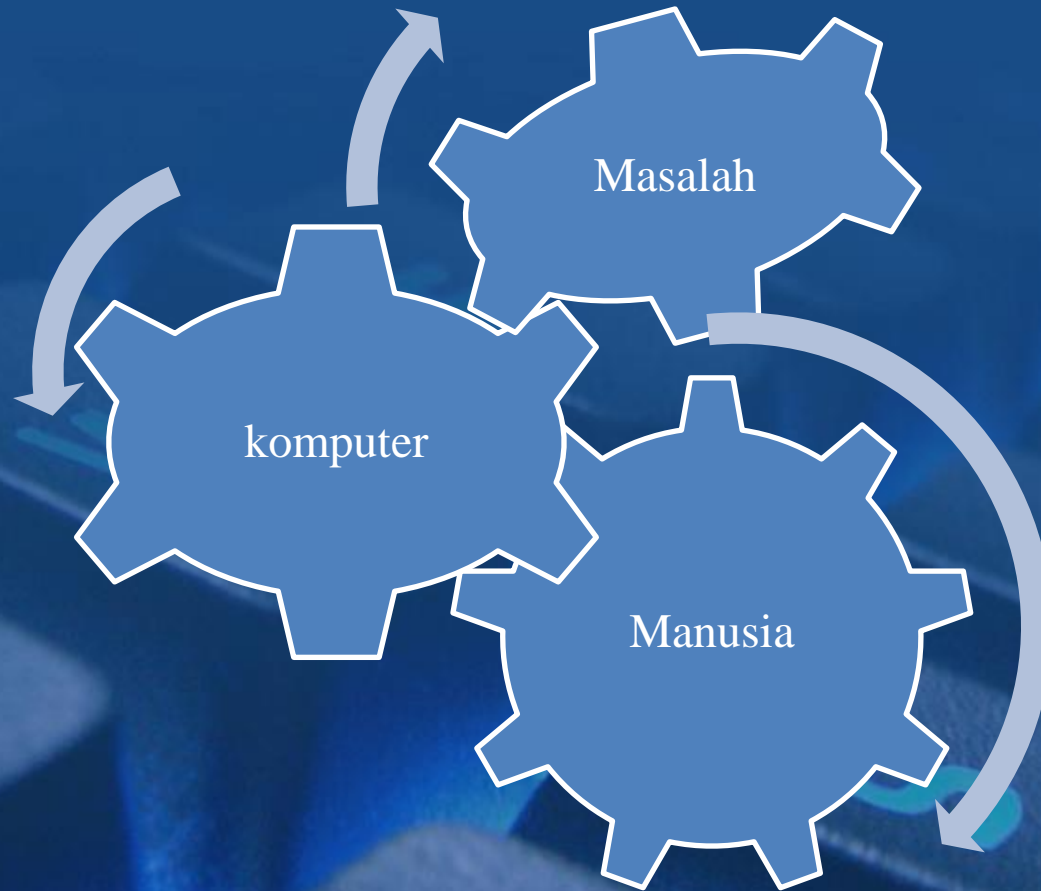
# Tujuan Pembelajaran :

1. Mahasiswa memahami cara kerja serta proses yang terjadi pada sebuah Compiler
2. Mahasiswa memahami konsep pembuatan sebuah Compiler
3. Mahasiswa mengetahui bagaimana sebuah bahasa pemrograman dapat diterjemahkan oleh mesin.

# Materi Pembelajaran

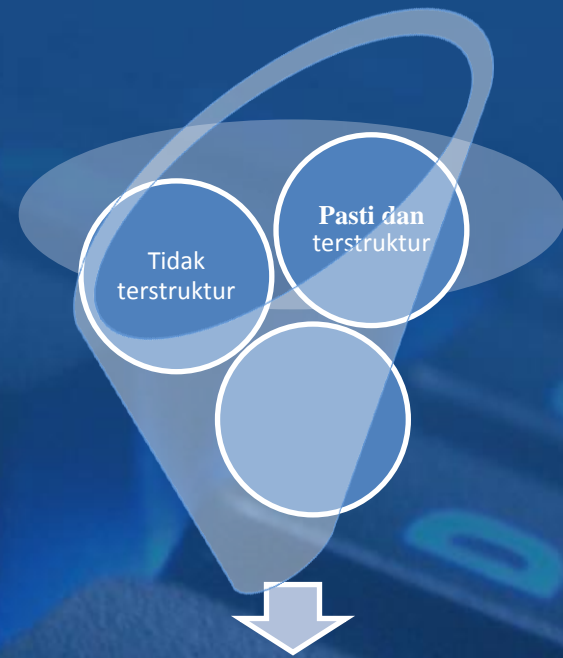
1. Bentuk-bentuk karakter dan kelas Grammar.
2. Ekspresi Regular dan Automata
3. Analisa Leksikal sebagai tahap awal kompilasi
4. Analisa Sintaks, bentuk-bentuk derivasi serta implementasi parsing.
5. Analisa Semantik dan tahapan Sintesa.
6. Penanganan kesalahan kompilasi dan fungsi tabel informasi.

# Bahasa



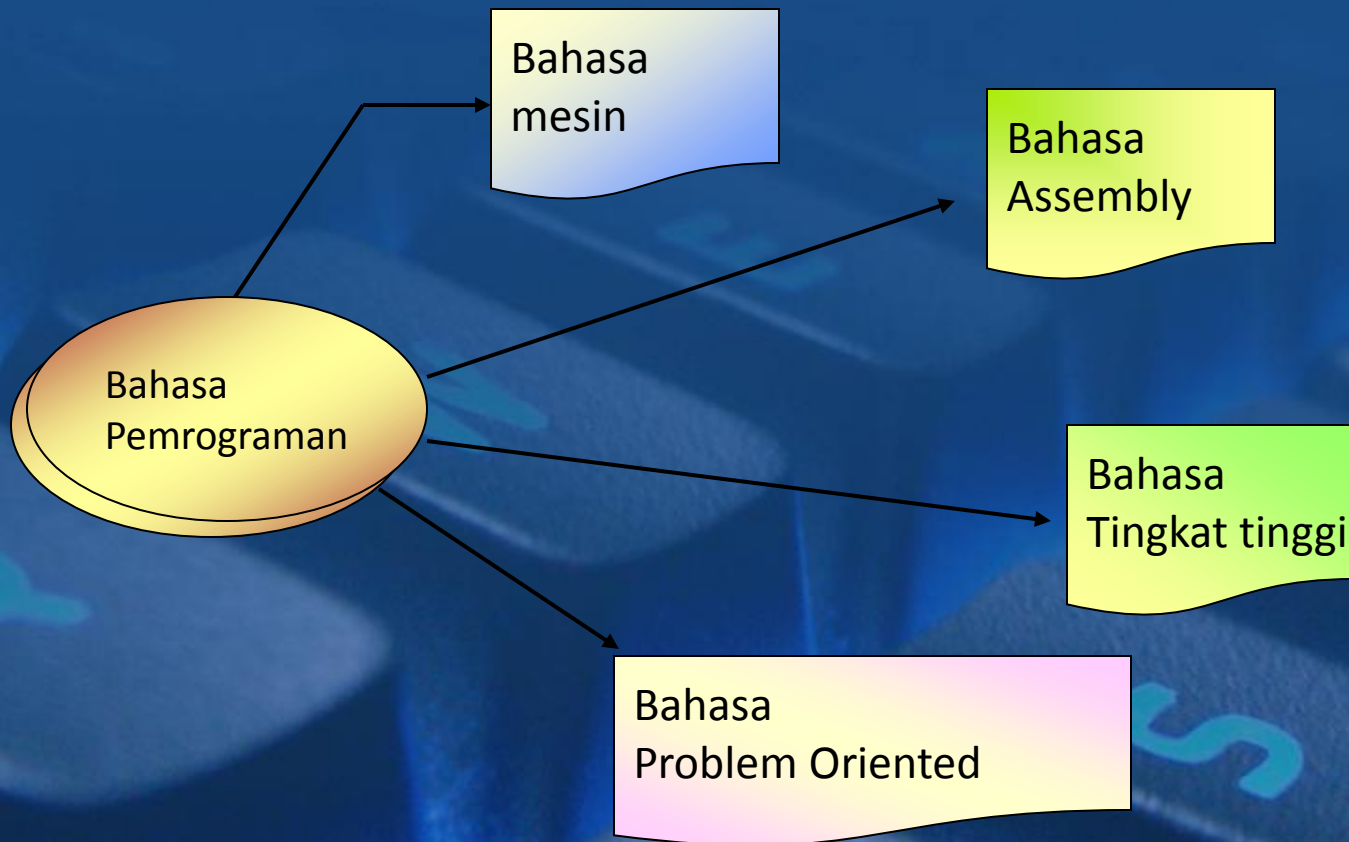
# Manusia-Masalah-Komputer

- ✓ Media interaksi secara efektif
- ✓ Media penyampaian gagasan dan pemikiran
- ✓ Menjembatani antara pemikiran manusia dengan komputer.
- ✓ Pemikiran manusia yg sering tidak terstruktur
- ✓ Komputer memerlukan hal yang pasti.
- ✓ Bahasa harus memiliki konstruksi (terminologi & elemen) = bahasa tingkat tinggi
- ✓ Komputer digital menerima bahasa tingkat rendah (0, 1) yg sulit dipahami manusia.



Bahasa yang bagaimana?

# Bahasa Pemrograman



**Bahasa mesin** merupakan bentuk terendah dari bahasa komputer. Instruksi direpresentasikan dalam kode numerik.

**Bahasa tingkat tinggi** (user oriented) lebih banyak memberikan fungsi kontrol program, nested, block, dan prosedur.co : Pascal

Program  
Language

**Bahasa Assembly** merupakan bentuk simbolik dari bahasa mesin. Kode misalnya ADD, MUL, dsb

Bahasa problem oriented sering juga dimasukkan sebagai bahasa tingkat tinggi, misalnya SQL, Myob, COGO (trmasuk bhs tingkat tinggi)

# Keuntungan Bahasa Tingkat Tinggi :

1. Kemudahan untuk dipelajari,tdk perlu latar belakang hardware.
2. Lebih mendekati permasalahan yang akan diselesaikan.
3. Memberikan banyak pilihan struktur kontrol (kondisional,looping,struktur block(begin-end),nested statement).
4. Program lebih mudah di-debug
5. Kemampuan struktur data yg lebih baik.
6. Deskripsi Modular

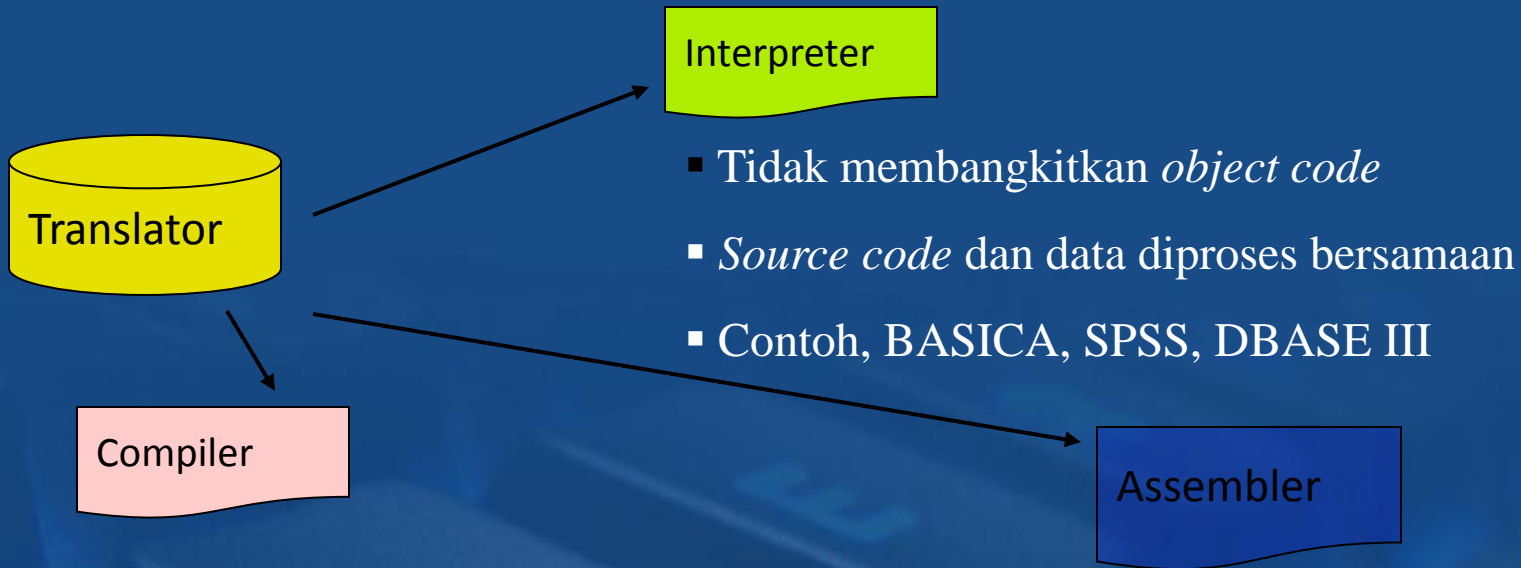


# Translator

- ✓ Agar dapat dieksekusi, program dalam bahasa tinggi harus ditranslasikan ke dalam bahasa mesin.

Translator melakukan perubahan perubahan source code / source program ke dalam target code / object code (bahasa mesin pada suatu komputer). Co:/ .pas menjadi .obj

Assembler, kompilator, Interpreter termasuk jenis dari translator.



Interpreter

- Tidak membangkitkan *object code*
- *Source code* dan data diproses bersamaan
- Contoh, BASICA, SPSS, DBASE III

Translator

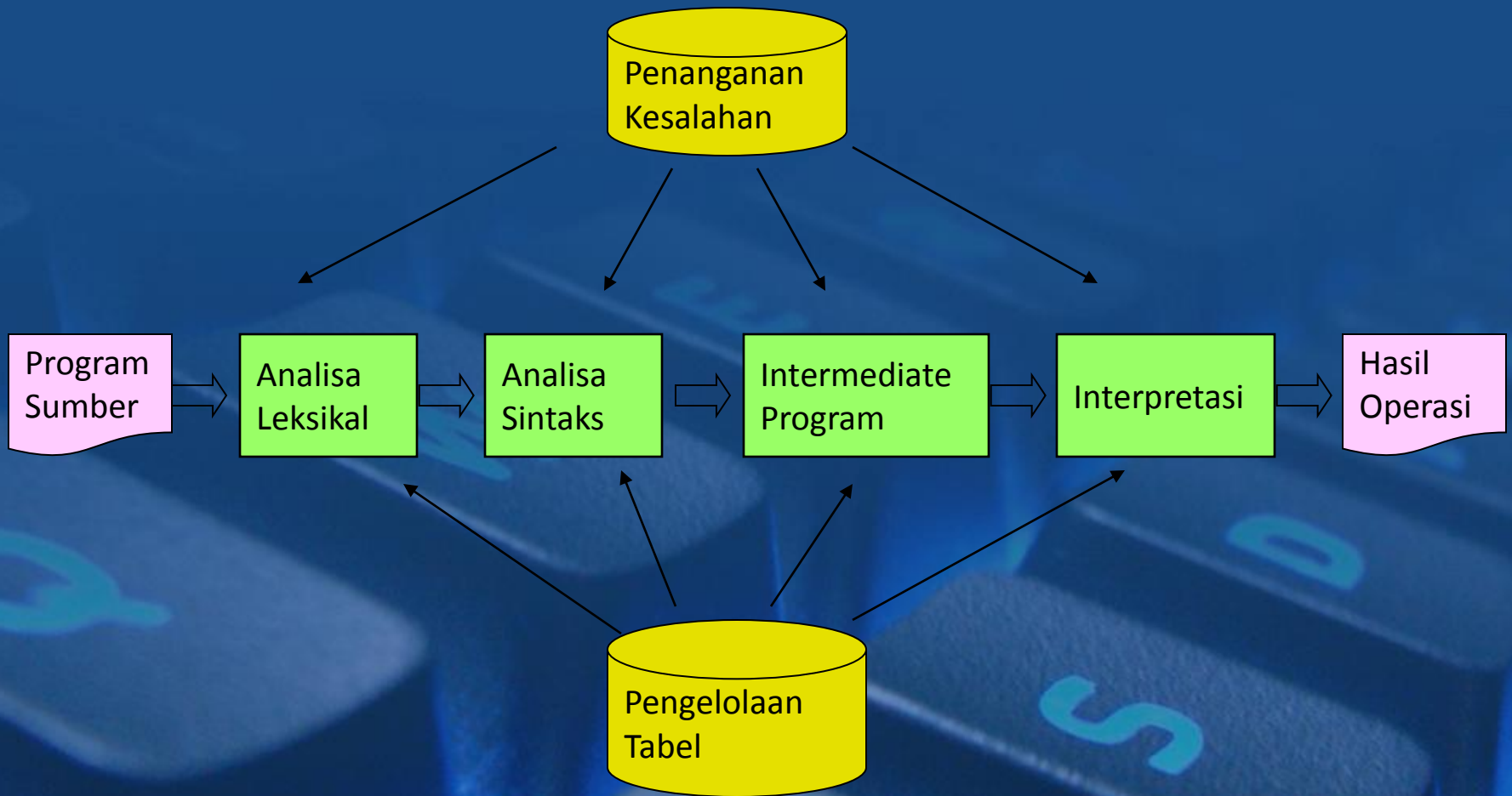
Compiler

Assembler

- *Source Code* adalah bahasa tingkat tinggi.
- *Object Code* adalah bahasa mesin atau assembly
- *Source code* dan data diproses tidak bersamaan
- Contoh, PASCAL, C

- *Source Code* adalah bahasa Assembly
- *Object Code* adalah bahasa mesin
- Contoh, Turbo Assembler

# Interpreter

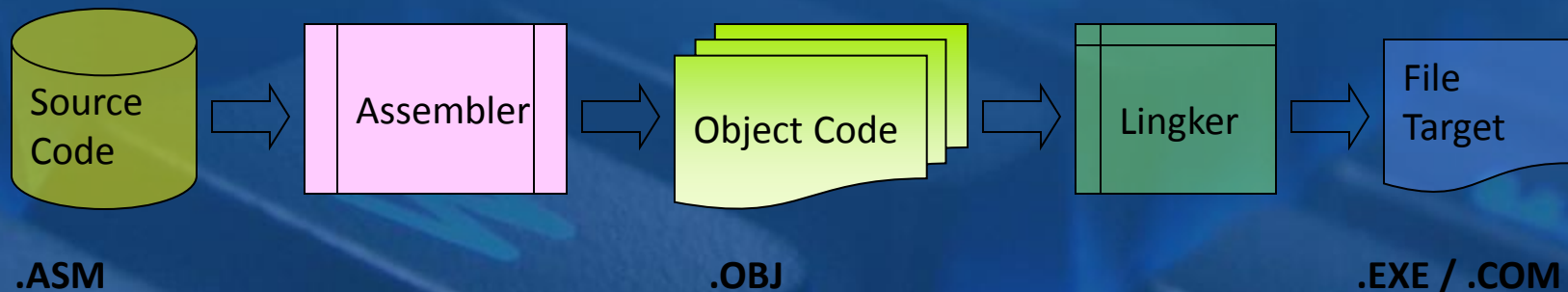


## ..... interpreter

Interpreter tidak membangkitkan object code, translasi hanya dalam bentuk internal. Contoh interpreter: BASIC, LISP, SMALLTALK (Matlab)

Source dan data diproses pada saat yang sama.

# Assembler



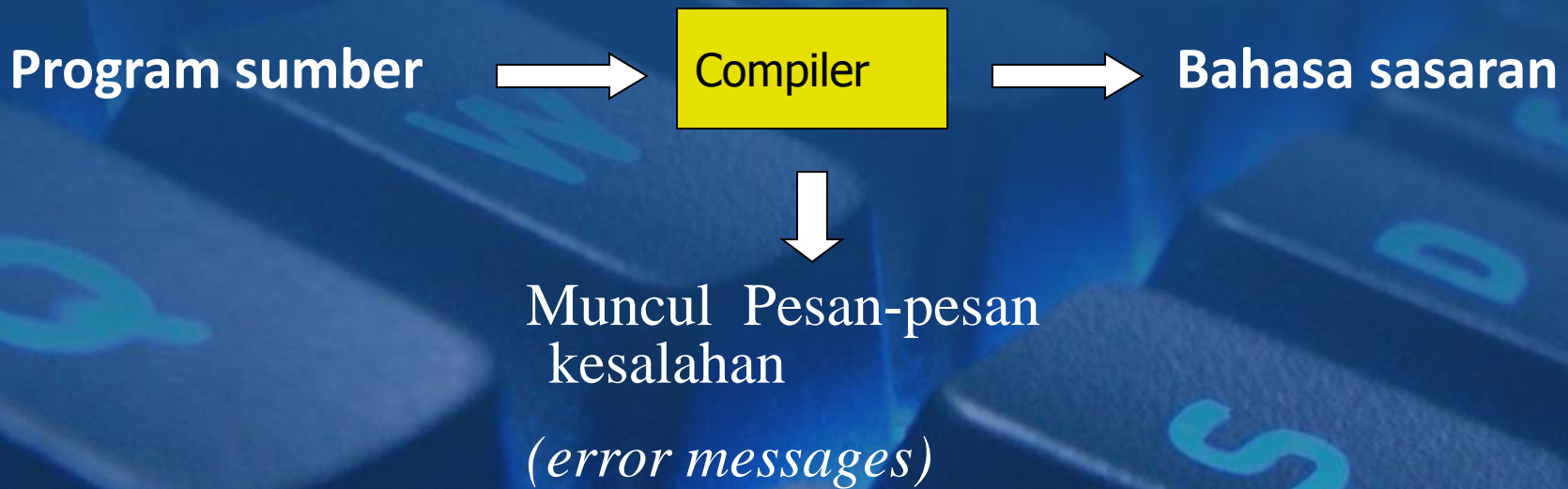
Proses Sebuah Kompilasi pada Bahasa Assembler

- Source Code adalah bahasa Assembler, Object Code adalah bahasa mesin
- Object Code dapat berupa file object (.OBJ), file .EXE, atau file .COM
- Contoh : Turbo Assembler (dari IBM) dan Macro Assembler (dari Microsoft)

# Compiler / Kompilator

Definisi : Kompilator (compiler) adalah sebuah program yang membaca suatu program yang ditulis dalam suatu bahasa sumber (*source language*) dan menterjemahkannya kedalam suatu bahasa sasaran (*target language*). Dari bahasa tingk tinggi ke bahasa mesin atau assembly. Source code dan data diproses pada saat yang berbeda.

Proses kompilasi dapat digambarkan melalui sebuah blok diagram sebagai berikut :



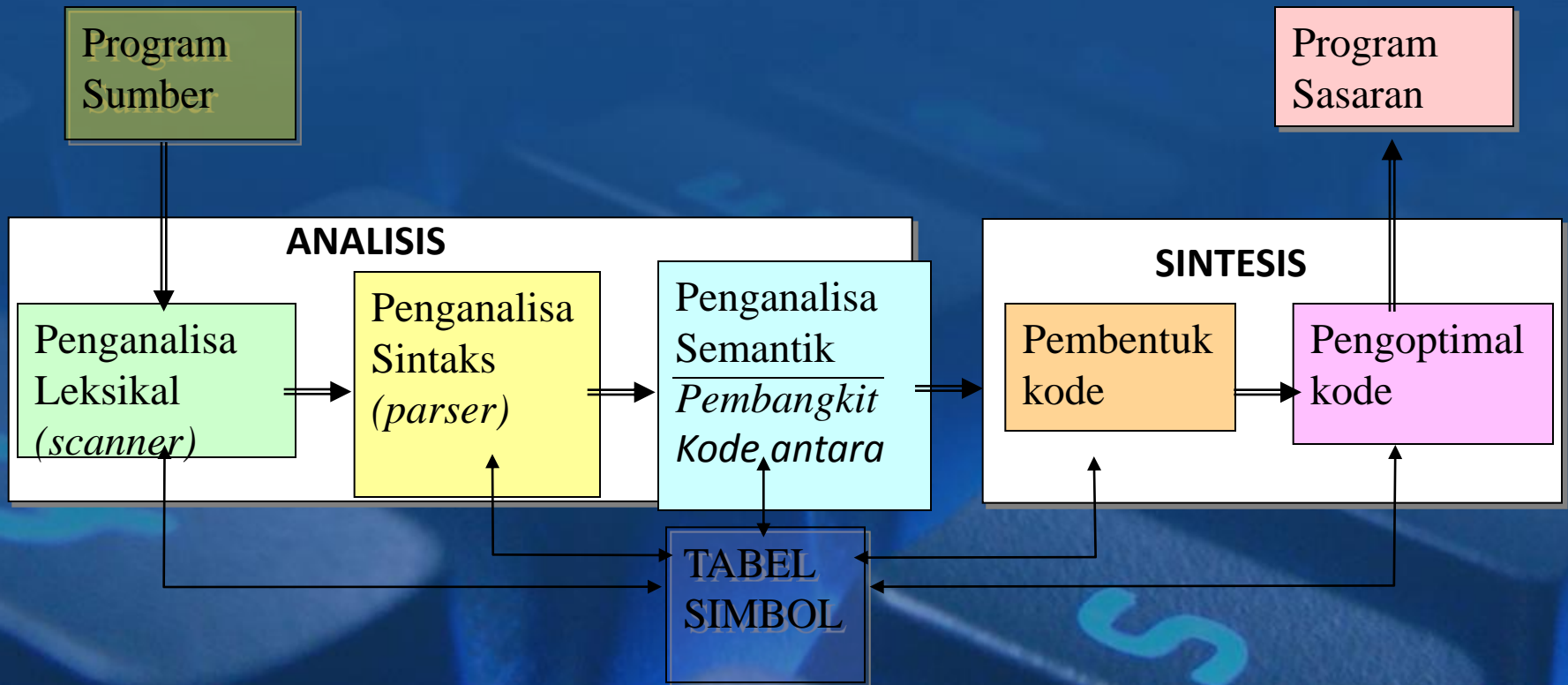
# Model Kompilator

Tugas Pokok dari Kompilator:

1. Fungsi Analisis ; biasa disebut front end. Tugasnya melakukan dekomposisi (pemecahan) program sumber menjadi bagian2 dasarnya.
2. Fungsi Sintesis ; biasa disebut back end. Tugasnya melakukan pembangkitan dan optimasi program objek.



# Blok Diagram



Bagan pokok proses kompilasi

# Keterangan Model Kompilator :

- ❑ **Program Sumber** ditulis dalam bahasa sumber, misal Pascal, Assembler, dsb
- ❑ **Program Sasaran** dapat berupa bahasa pemrograman lain atau bahasa mesin pada suatu komputer
- ❑ **Scanner** : Memecah program sumber menjadi besaran leksik/token
- ❑ **Parser** : Memeriksa kebenaran dan urutan kemunculan token
- ❑ **Penganalisa semantik** : Melakukan analisa semantik, biasanya dalam realisasi akan digabungkan Dengan *intermediate code generator* (bagian yang berfungsi membangkitkan kode antara)
- ❑ **Pembentuk Kode** : Membangkitkan kode objek
- ❑ **Pengoptimal Kode** : Memperkecil hasil dan mempercepat proses
- ❑ **Tabel** : Menyimpan semua informasi yang berhubungan dengan proses kompilasi